

(11)特許出願公開番号

特開2001-150739

(P2001-150739A)

(43)公開日 平成13年6月5日(2001.6.5)

(51) Int.Cl.⁷

識別記号

FI

テーマト* (参考)

B 4 1 J 5/30

B 4 1 J 5/30

Z 2C087

G O 6 F 3/12

G 0 6 F 3/12

H 5 B 0 2 1

9A001

審査請求 未請求 請求項の数8 O.L (全 15 頁)

(21)出願番号

特願平11-337158

(22) 出願日

平成11年11月29日(1999. 11. 29)

(71)出國人 591044164

株式会社沖データ

東京都港区芝浦四丁目11番地22号

(72)発明者 山本 聡

東京都港区芝浦四丁目11番地22号 株式会社
沖デ一夕内

(72)発明者 石川 修

東京都港区芝浦四丁目11番地22号 株式会社
沖デ一夕内

(74) 代理人 100082050

弁理士 佐藤 幸男 (外1名)

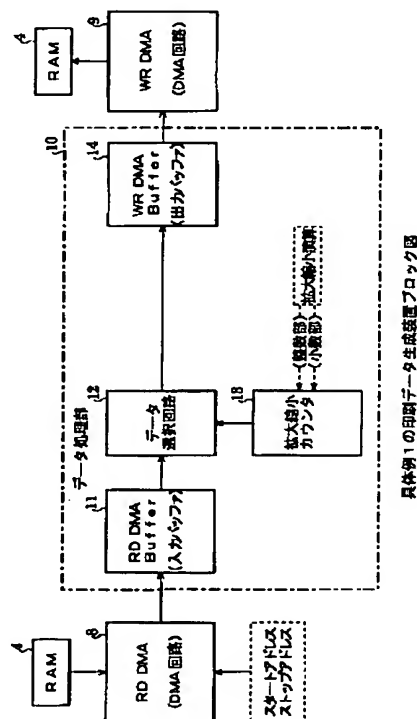
最終頁に続く

(54) 【発明の名称】 印刷データ生成装置

(57) 【要約】

【解決手段】 入力バッファ１１は、入力データを保持する。拡大縮小カウンタ１３は、入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、入力バッファ１１に保持されたいずれかのデータを選択する選択信号を生成する。データ選択回路１２は、入力バッファ１１に保持されたデータを選択して、出力バッファ１４に出力する。データ処理部１０は、ハードウェアによって構成する。

【効果】 拡大縮小処理部や二値化処理部をハードウェアにより構成するので、CPUに対する負荷をかけずに印刷データ生成処理の高速化が図れる。



【特許請求の範囲】

【請求項1】 入力データを保持する入力バッファと、前記入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、前記入力バッファに保持された1画素分のデータを選択する選択信号を生成して出力する拡大縮小カウンタと、前記拡大縮小カウンタの出力する選択信号により前記入力バッファに保持された1画素分のデータを所定の順に選択して、前記入力データを拡大縮小した出力データを生成するデータ選択回路とを備えたことを特徴とする印刷データ生成装置。

【請求項2】 請求項1に記載の印刷データ生成装置において、拡大縮小カウンタは、入力データのライン長を出力データのライン長で除算した結果の整数部を保持する整数部倍率レジスタと、入力データのライン長を出力データのライン長で除算した結果の小数部を保持する小数部倍率レジスタと、入力データの先頭に配置された1画素分のデータを読み出すアドレスを初期値として、1画素分のデータ読み出し毎に、前記整数部倍率レジスタの保持する整数部を累積加算して出力する整数部倍率カウンタと、ゼロを初期値として、1画素分のデータ読み出し毎に、前記小数部倍率レジスタの保持する小数部を累積加算して、この累積加算値が1を越えたとき、前記整数部に1を加算するためのデータを出力する小数部倍率カウンタとを備えたことを特徴とする印刷データ生成装置。

【請求項3】 階調データを含む入力データを保持する入力バッファと、前記入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、前記入力バッファに保持された1画素分のデータを選択する選択信号を生成して出力する拡大縮小カウンタと、前記拡大縮小カウンタの出力する選択信号により前記入力バッファに保持された1画素分のデータを所定の順に選択して、前記入力データを拡大縮小した出力データを生成するデータ選択回路と、1画素分のデータ読み出し毎に、所定の順にディザデータを構成する閾値を読み出して出力するディザデータ回路と、前記閾値を受け入れて、前記選択回路の出力を二値化する二値化回路とを備えたことを特徴とする印刷データ生成装置。

【請求項4】 請求項3に記載の印刷データ生成装置において、ディザデータ回路は、外部メモリからディザデータを分割して受け入れるダイレクトメモリアクセス回路を備えたことを特徴とする印刷データ生成装置。

【請求項5】 請求項1に記載の印刷データ生成装置において、

入力バッファの入力側に、入力データの先頭に含まれており、出力データに含めることが不要な無効データの数を保持して、入力データの入力バッファへの書き込み開始位置を、前記無効なデータ数だけシフトさせるように制御するオフセット制御回路を設けたことを特徴とする印刷データ生成装置。

【請求項6】 請求項1に記載の印刷データ生成装置において、

出力バッファの入力側に、出力データに含めることが不要な無効データの数を保持して、出力データの出力バッファへの書き込み開始位置を、前記無効なデータ数だけシフトさせるように制御するオフセット制御回路を設けたことを特徴とする印刷データ生成装置。

【請求項7】 請求項1に記載の印刷データ生成装置において、

出力データを保持する出力バッファと、予め設定された規定データ長を保持し、前記出力バッファに書き込まれる出力データ長が前記規定データ長を越えた場合には、出力データの前記規定データ長を越えた部分を破棄し、前記出力バッファに書き込まれる出力データ長が前記規定データ長に満たない場合には、出力データの前記規定データ長に満たない部分を補充するように前記出力バッファへの出力データの書き込みを制御するライン長補正回路を備えたことを特徴とする印刷データ生成装置。

【請求項8】 請求項1に記載の印刷データ生成装置において、

外部メモリから入力バッファへ、入力データを転送するDMA回路と、

入力データを縮小した出力データを生成する場合に、前記DMA回路が前記外部メモリから入力データを、縮小率に応じた間隔でスキップして読み出すように制御するスキップ制御回路を設けたことを特徴とする印刷データ生成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、階調性を持つイメージデータの拡大縮小や二値化処理を行う印刷データ生成装置に関する。

【0002】

【従来の技術】コンピュータで生成されたイメージデータをプリンタで印刷する場合には、予めプリンタの機能に応じたデータの変換処理が行われる。例えば、入力した印刷用イメージデータの解像度をプリンタの印刷解像度に合わせるために、イメージデータの拡大あるいは縮小処理が行われる。拡大処理の場合には、イメージデータを1ライン分ずつバッファメモリに記憶し、各画素のデータを拡大倍率分だけコピーして画素数を増やす。同様にしてライン数もコピーによって増やす。一方、縮小

処理の場合には、1ライン分のイメージデータをバッファメモリに格納し、画素の間引き処理を行って画素数を減らし、その結果を出力する。出願するライン数も減らす。

【0003】また、多くのプリンタは、黒ドットを印刷するかしないかという二値制御によって画像を表現する。従って、各画素が濃淡を表す情報を持つ階調データの場合には、ディザマトリクスを用いた二値化処理が行われる。こうした処理では、各画素の階調データが1つずつ読み出され、ディザマトリクスを構成する閾値と比較され、二値化が行われる。

【0004】

【発明が解決しようとする課題】ところで、上記のような従来の技術には、次のような解決すべき課題があった。プリンタは、コンピュータ等の上位装置から印刷データを受け入れると、上記のような拡大縮小処理や二値化処理を実行して印刷データを生成する。プリンタのCPU（中央処理装置）は、印刷データをライン単位あるいは画素単位でバッファメモリに呼び出し、拡大縮小あるいは二値化のための演算処理を行って、その結果をプリントエンジンに向けて出力する。

【0005】しかしながら、この種の処理は、1画素分のデータを単位とした演算処理の繰り返しが多く含まれ、高い解像度のイメージデータの場合、その演算処理量は膨大になる。しかも、階調データ1画素分は例えば8ビット程度のデータであり、バス幅が32ビットあるいはそれ以上の広さを持つCPUであっても、そのバス幅を有効に利用した効率的な演算処理は難しい。即ち、バス幅の広いCPU本来の高速演算処理能力を有効に発揮することができず、バッファメモリへの頻繁な中間データアクセスのために演算処理時間が長くなってしまいうという問題があった。また、こうした印刷データ生成処理に大きな負荷がかかることによって、CPUによるプリンタの他の部分の制御機能を圧迫してしまうという問題もあった。

【0006】

【課題を解決するための手段】本発明は以上の点を解決するため次の構成を採用する。

〈構成1〉入力データを保持する入力バッファと、上記入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、上記入力バッファに保持された1画素分のデータを選択する選択信号を生成して出力する拡大縮小カウンタと、上記拡大縮小カウンタの出力する選択信号により上記入力バッファに保持された1画素分のデータを所定の順に選択して、上記入力データを拡大縮小した出力データを生成するデータ選択回路とを備えたことを特徴とする印刷データ生成装置。

【0007】〈構成2〉構成1に記載の印刷データ生成装置において、拡大縮小カウンタは、入力データのライ

ン長を出力データのライン長で除算した結果の整数部を保持する整数部倍率レジスタと、入力データのライン長を出力データのライン長で除算した結果の小数部を保持する小数部倍率レジスタと、入力データの先頭に配置された1画素分のデータを読み出すアドレスを初期値として、1画素分のデータ読み出し毎に、上記整数部倍率レジスタの保持する整数部を累積加算して出力する整数部倍率カウンタと、ゼロを初期値として、1画素分のデータ読み出し毎に、上記小数部倍率レジスタの保持する小数部を累積加算して、この累積加算値が1を越えたとき、上記整数部に1を加算するためのデータを出力する小数部倍率カウンタとを備えたことを特徴とする印刷データ生成装置。

【0008】〈構成3〉階調データを含む入力データを保持する入力バッファと、上記入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、上記入力バッファに保持された1画素分のデータを選択する選択信号を生成して出力する拡大縮小カウンタと、上記拡大縮小カウンタの出力する選択信号により上記入力バッファに保持された1画素分のデータを所定の順に選択して、上記入力データを拡大縮小した出力データを生成するデータ選択回路と、1画素分のデータ読み出し毎に、所定の順にディザデータを構成する閾値を読み出して出力するディザデータ回路と、上記閾値を受け入れて、上記選択回路の出力を二値化する二値化回路とを備えたことを特徴とする印刷データ生成装置。

【0009】〈構成4〉構成3に記載の印刷データ生成装置において、ディザデータ回路は、外部メモリからディザデータを分割して受け入れるダイレクトメモリアクセス回路を備えたことを特徴とする印刷データ生成装置。

【0010】〈構成5〉構成1に記載の印刷データ生成装置において、入力バッファの入力側に、入力データの先頭に含まれており、出力データに含めることが不要な無効データの数を保持して、入力データの入力バッファへの書き込み開始位置を、上記無効なデータ数だけシフトさせるように制御するオフセット制御回路を設けたことを特徴とする印刷データ生成装置。

【0011】〈構成6〉構成1に記載の印刷データ生成装置において、出力バッファの入力側に、出力データに含めることが不要な無効データの数を保持して、出力データの出力バッファへの書き込み開始位置を、上記無効なデータ数だけシフトさせるように制御するオフセット制御回路を設けたことを特徴とする印刷データ生成装置。

【0012】〈構成7〉構成1に記載の印刷データ生成装置において、出力データを保持する出力バッファと、予め設定された規定データ長を保持し、上記出力バッファに書き込まれる出力データ長が上記規定データ長を越

えた場合には、出力データの上記規定データ長を越えた部分を破棄し、上記出力バッファに書き込まれる出力データ長が上記規定データ長に満たない場合には、出力データの上記規定データ長に満たない部分を補充するように上記出力バッファへの出力データの書き込みを制御するライン長補正回路を備えたことを特徴とする印刷データ生成装置。

【0013】〈構成8〉構成1に記載の印刷データ生成装置において、外部メモリから入力バッファへ、入力データを転送するDMA回路と、入力データを縮小した出力データを生成する場合に、上記DMA回路が上記外部メモリから入力データを、縮小率に応じた間隔でスキップして読み出すように制御するスキップ制御回路を設けたことを特徴とする印刷データ生成装置。

【0014】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

〈具体例1〉本発明では、CPUの負担を軽減するために、画素単位の演算処理を可能な限りハードウェアを利用して行う。図1は、具体例1の印刷データ生成装置を示すブロック図である。図の装置は、RAM（ランダムアクセスメモリ）4からDMA回路（RD DMA（ダイレクトメモリアクセスコントローラ））8を用いて読み出した入力データをデータ処理部10において拡大縮小処理し、DMA回路9を用いてRAM4に書き込むよう構成されている。データ処理部10は、入力バッファ11、データ選択回路12、拡大縮小カウンタ13及び出力バッファ14により構成される。

【0015】この回路の具体的な動作を説明する前に、まず装置全体の構成を説明する。図2は、印刷装置の全体構成を示すブロック図である。図の装置は、CPU1がバスライン2を介して装置各部を制御するように構成されている。バスライン2には、ROM（リードオンリメモリ）3、RAM4、ホストインタフェース制御部5、印刷制御部6及び印刷データ生成装置7が接続されている。印刷データ生成装置7には、読み出し用のDMA回路（RD DMA）8と、データ処理部10と、書き込み用のDMA回路（WR DMA）9とが設けられている。

【0016】ROM3は、CPU1の動作プログラム等を格納したメモリである。RAM4は、CPU1による動作パラメータや処理対象となる入力データあるいは印刷データを格納するためのメモリである。ホストインタフェース制御部5は、図示しない上位装置とプリンタとの通信制御を行う部分で、このホストインタフェース制御部5を通じて入力データが上位装置から入力し、RAM4に記憶される。印刷制御部6は、図示しないプリントエンジンを制御し、RAM4に記憶された印刷データの印刷を制御する部分である。

【0017】印刷データ生成装置7は、RAM4にホス

トインタフェース制御部5を通じて受け入れられた入力データの拡大縮小処理や二値化処理を行って、印刷データを生成する部分である。読み出し用のDMA回路8は、RAM4から入力データを順番に読み出してデータ処理部10に供給する回路である。書き込み用のDMA回路9はデータ処理部10で処理の終了した印刷データをRAM4に順番に書き込む動作を行う回路である。

【0018】本発明においては、この印刷データ生成装置7のデータ処理部10がハードウェアにより構成され、拡大縮小処理や二値化処理をCPU1の演算処理によらずに実行する。読み出し用のDMA回路8と書き込み用のDMA回路9とは、いずれも、従来一般に知られたダイレクトメモリアクセスコントローラで、ハードウェアにより構成される。

【0019】図3に、データ処理部の機能説明図を示す。（a）は、拡大処理機能の内容を図示したものである。（b）は、縮小処理機能の内容を図示したものである。（c）は、二値化処理機能の内容を図示したものである。図の（a）に示すように、例えば1画素を8ビットのデジタルデータで表現する入力データD1を2倍に拡大する場合には、1画素分の入力データを出力データD0に2回ずつコピーして出力する。

【0020】（b）に示す縮小処理の場合には、入力データD1を1画素分ずつ1つおきに取り出して、出力データD0とする。即ち、間引き処理によって縮小を行う。（c）に示す二値化処理では、後で説明するようなディザパターンデータD3に含まれる閾値を所定の順番に読み出して、入力データD1の各画素の画素値と比較する。こうして二値化されたデータを印刷データとする。

【0021】再び図1に戻って、図1に示すデータ処理部10を含めた印刷データ生成装置の説明をする。読み出し用のDMA回路8は、図2に示したCPU1から入力データのスタートアドレスとストップアドレスとを受け入れて、RAM4から該当する入力データを自動的に読み出す処理を行うよく知られた回路である。入力バッファ11には、この入力データが一定量ずつ順番に書き込まれる。少なくとも1ライン分のデータが書き込めることが望ましいが、2分の1ライン分ずつでも数ライン分ずつでもかまわない。入力バッファ11に保持された入力データの拡大縮小処理が終了すると、DMA回路8に対し次の入力データの転送を要求するといった手順によって、入力バッファ11に入力データが順番に送り込まれる。

【0022】データ選択回路12は、拡大縮小カウンタ13から出力される選択信号を受け入れて、入力バッファ11に記憶された入力データのうちの該当する1画素分のデータを読み出し、これを出力バッファ14に転送する処理を繰り返し行う回路である。例えば、入力バッファ11が、1ライン分の階調データを保持することが

できるメモリとしたとき、データ選択回路12は、拡大縮小カウンタ13から出力される選択信号を入力バッファ11のアドレス信号として供給し、入力バッファ11から読み出された1画素分のデータを出力バッファ14に転送するバスラインやゲート回路により構成する。

【0023】出力バッファ14は、データ選択回路12を通じて受け入れたデータを受け入れた順に記憶するメモリから構成される。そのメモリ容量は、出力画像の1ライン分以上あることが好ましいが、入力バッファ11と同様に任意の容量で差し支えない。書き込み用DMA回路9は、例えば、入力バッファ11に保持された入力データの処理が終了し、入力バッファ11に新たな入力データが書き込まれるタイミングで動作し、出力バッファ14に記憶された印刷データをRAM4に転送する制御を行う回路である。こうして転送された印刷データはRAM4の所定の領域に順番に書き込まれ、全体として、例えば1ページ分の印刷データとなる。

【0024】図4に、拡大縮小カウンタのブロック図を示す。上記拡大縮小カウンタ13は、例えばこの図に示すように、整数部倍率レジスタ21、小数部倍率レジスタ22、整数部倍率加算器23、整数部倍率カウンタ24、小数部倍率加算器25、小数部倍率カウンタ26により構成される。

【0025】拡大縮小処理を行う場合、予め、図2に示すCPU1は、入力データのライン長を出力データのライン長で除算した結果を求める。拡大倍率が2倍なら、“0.5”、2分の1なら“2”である。そして、その整数部例えば“0”を整数部倍率レジスタ21に書き込む。また、小数部例えば“5”を小数部倍率レジスタ22に書き込む。このような処理を含む準備を行った後、上記データ処理部10の動作が開始される。

【0026】整数部倍率加算器23は、整数部倍率レジスタ21に記憶された内容と、小数部倍率カウンタ26から出力されたキャリーと、整数部倍率カウンタ24の出力とを加算して、次のタイミングで整数部倍率カウンタ24に記憶させる機能を持つ。小数部倍率加算器25は、小数部倍率レジスタ22の出力と、小数部倍率カウンタ26の出力とを加算して、次のタイミングで小数部倍率カウンタ26に記憶させる機能を持つ。

【0027】整数部倍率カウンタ24も小数部倍率カウンタ26も、初期値は“0”である。そして、1画素分ずつ入力データを処理するための制御クロックにタイミングを合わせて、この演算処理を実行する。即ち、一番最初の画素を処理する場合の出力は、整数部倍率カウンタ24に記憶された初期値の“0”である。そして、次の制御クロックが入力すると、整数部倍率加算器23の加算結果が整数部倍率カウンタ24に記憶される。そして、これが次の出力となる。なお、この整数部倍率カウンタ24の出力する選択信号Sは、入力バッファ11に記憶された例えば1ライン分の入力データ中の特定の画

素を読み出すためのアドレス信号になる。

【0028】図5は、拡大縮小動作の説明図である。この図を用いて、上記拡大縮小カウンタの動作原理や具体的な出力の内容を説明する。図5は、縮小処理の場合の出力データの画素位置と、整数部倍率レジスタ21の内容（整数部＝）と、小数部倍率レジスタ22の内容（小数部＝）と、小数部倍率カウンタ26の内容Pregと、整数部倍率カウンタ24の内容Regとの関係を示したものである。（a）は縮小処理、（b）は拡大処理を示す。

【0029】ここで、例えば2分の1縮小処理を実行する場合を考える。このとき、入力データライン長を出力データライン長で除した値は“2”となる。整数部倍率レジスタ21には、この除算結果の“2”が書き込まれる。小数部倍率レジスタ22には、小数部即ち“0.0”が書き込まれる。小数部倍率カウンタ26の内容Regは初期値“0.0”である。整数部倍率カウンタ24の内容Pregは初期値“0”である。

【0030】まず、一番最初の画素を処理するタイミングでは、整数部倍率カウンタ24に初期値“0”が格納されているから、その値が選択信号Sとして出力される。入力バッファ11に1ライン分の入力データが保持されていれば、アドレスが“0”即ち先頭の画素のデータが出力バッファ14の先頭の画素として書き込まれる。次の制御クロックが入力すると、整数部倍率レジスタ21に記憶された“2”と、整数部倍率カウンタ24に記憶された“0”とが加算される。

【0031】小数部倍率カウンタ26には、初期値“0”が記憶され、小数部倍率レジスタ22には、“0.0”が書き込まれている。従って、小数部倍率加算器25の加算結果は“0”であり、これが小数部倍率カウンタ26に再び書き込まれる。このため、小数部倍率カウンタ26はキャリーを発生しない。その結果、整数部倍率カウンタ24には、整数部倍率レジスタ21に記憶された“2”と整数部倍率カウンタ24に記憶された“0”の加算結果2が書き込まれる。こうして、次のタイミングで“2”という選択信号Sが出力される。

【0032】このように、選択信号が“0”の次は“2”となったため、入力データのアドレスが“2”の画素が読み出されて出力バッファ14に書き込まれる。同様に、次の制御クロックが入力すると、整数部倍率カウンタ24の内容が“2”で、整数部倍率レジスタ21の内容の“2”と加算されるから、選択信号Sは“4”となる。即ち、図5（a）の2分の1倍縮小と表示したPregの列に記載したように、選択信号Sは、“0”，“2”，“4”，“6”，“8”，…というように出力される。その結果、入力データ中の画素が1つおきに選択されて出力バッファ14に記憶される。上記拡大縮小カウンタ13は、このようにして入力バッファ11に記憶された入力データのいずれかを選択し、拡大

縮小処理した結果を出力バッファ14に転送する機能を持つ。

【0033】次に、拡大処理について説明を行う。例えば入力データを2倍に拡大する場合には、入力データのライン長を出力データのライン長で除し、“0.5”という結果を得る。整数部倍率レジスタ21には“0”というデータが書き込まれる。一方、小数部倍率レジスタ22には“0.5”という結果が書き込まれる。なお、小数部倍率レジスタ22の精度を小数点以下2桁とし、図5(b)の例では、小数部倍率レジスタ22に“0.50”という数値を記憶する。除算結果に端数があれば、端数を切り上げや切り捨て等により処理して小数部を求めれば良い。

【0034】ここで、最初の制御クロックが入力した場合には、2分の1倍縮小の際と同様に、整数部倍率カウンタ24に記憶された初期値である“0”が選択信号Sとして出力される。従って、最初の画素のデータが出力バッファ14の先頭の画素として出力されることは変わらない。次の制御クロックが入力すると、整数部倍率カウンタ24には、整数部倍率レジスタ21の出力と、整数部倍率カウンタ24にこれまで記憶されていた値の加算値が入力する。この場合、両方とも“0”であるから、整数部倍率カウンタ24には“0”がそのまま記憶される。

【0035】小数部倍率カウンタ26には、小数部倍率レジスタに記憶された“0.5”という値が記憶される。小数部倍率カウンタ26は、ここに記憶されたデータが“1”に達するとキャリーを出力する。小数部倍率カウンタ26の内容が“0”や“0.5”の場合にはキャリーが出力されないから、整数部倍率カウンタ24からは、今度も内容が“0”の選択信号Sが出力される。

【0036】更に、次の制御クロックが入力した場合を考える。このとき、小数部倍率カウンタ26には“0.5”が記憶されており、小数部倍率加算器25は小数部倍率レジスタ22の出力と小数部倍率カウンタ26の出力とを加算して“1.00”という結果を得る。これが小数部倍率カウンタ26に送り込まれると、小数部倍率カウンタ26は桁あふれのため、“00”のみを格納し、キャリーを整数部倍率加算器23に出力する。整数部倍率カウンタ24は、これまで“0”を記憶している。整数部倍率レジスタ21の内容は“0”である。整数部倍率加算器23は、これらの加算結果である“1”を整数部倍率カウンタ24に出力する。こうして、整数部倍率カウンタ24は、内容が“1”の選択信号Sを出力する。その結果、図5(b)に示すように、3回目には“1”という選択信号が出力される。

【0037】同様の処理を繰り返すと、選択信号は、“0”、“0”、“1”、“1”、“2”、“2”、“3”、“3”、…というように変化する。その結果、入力データの各画素が2回ずつ重複して出力バッファ1

4に転送され、1ライン分の入力データがライン方向に2倍に拡大される。このように、小数部倍率カウンタ26は“1”に満たない端数を累積加算して、その累積加算結果が“1”以上になるとキャリーを出力し、整数部倍率カウンタ24に“1”を加算する作用をする。これによって、任意の拡大倍率を指定した場合に必要な数の画素が所定のタイミングでコピーされて、該当する拡大倍率の印刷データを得ることができる。その他の倍率については、同様の手順で処理を行うため、具体的な説明は省略するが、図5(a)と(b)にその一例を図示した。

【0038】なお、上記の例で、入力データは、二値データでも階調データでもよい。このとき、入力データのビット数にかかわらず1画素分のデータを8ビットに揃えるとよい。単一のハードウェアにより入力データを画一的に処理するためである。この場合、1画素分の入力データが8ビット以下のビット数ならば、DMA回路8が、入力データをそのまま入力バッファ11の8ビット幅の記憶領域に順に書き込めば良い。一方、1画素分の入力データが8ビットを超える場合もあるならば、入力データを一律に正規化する回路を設ける。即ち、CPUが予め入力データのビット数を書き込んでおくレジスタを設け、出力データのビット数“8”をこのレジスタ値で除算した値と入力データとの積を演算して、自動的にデータの正規化をする乗算回路を付加すればよい。例えば、入力データが1画素当たり4ビットの階調データを8ビットに正規化するには、1画素毎にデータを17倍する処理を行う。

【0039】また、上記の例ではデータのライン方向(主走査方向)の拡大縮小処理のみを示したが、副走査方向の拡大縮小処理も同様にして行える。しかしながら、副走査方向の拡大処理は各ラインを拡大倍率分コピーすればよく、縮小処理は各ラインを間引きすればよいので、主走査方向の処理に比べて演算処理が簡単である。また、データ長も長いからCPUによる高速演算処理に適している。従って、上記の回路の出力を受け入れてCPUがソフトウェア処理することが好ましい。もちろん、上記の回路の入力側で、予め副走査方向の拡大縮小処理を済ませておくようにしてもよい。これにより、主走査方向も副走査方向もハードウェアで処理する場合に比べてハードウェア規模の拡大を防止し、装置の小型化や低コスト化を図ることができる。なお、イメージデータの拡大縮小処理はページ単位で行ってもよいし、また任意のサイズのオブジェクト単位で行ってもよい。

【0040】〈具体例1の効果〉以上のように、上記の印刷データ生成装置は、入力データのライン長を出力データのライン長で除算した結果の整数部と小数部とを受け入れて、加算器を用いてカウントし、入力バッファに記憶された入力データを順番に選択して出力データとするようにデータ処理部が動作するため、ハードウェアに

よる拡大縮小処理を連続的に自動的に実行することができる。従って、CPUに負荷を与えずにこの種の画像処理を高速に行うことが可能になる。

【0041】また、上記の装置では、RAM4に記憶された入力データを読み出し、最終的な処理結果をRAM4に書き込むといった処理を行うため、バスの占有率が少なく、CPUによる他の処理を妨げることがない。即ち、入力データ処理のためにRAM4を繰り返し頻繁にアクセスすることが不要になり、RAM4のトータルアクセス回数を十分に低減することができる。

【0042】〈具体例2〉具体例1では、RAM4に記憶された入力データから印刷データを生成するための処理にあたって、RAM4から入力データを読み出すための処理を1回で行い、バスラインの占有時間を短縮するようにした。ここで、上記のような拡大縮小処理の後、更にそのデータの二値化処理を行おうとすれば、再びRAM4に記憶された中間データを二値化処理して印刷データにする手順を実行しなければならない。これは従来のCPUによる処理と同様に、RAM4に対するアクセス回数を増加させることになる。そこで、この具体例では、1回の読み出しによって、データ処理部10が拡大縮小処理と同時に二値化処理も実行する。

【0043】図6には、具体例2のデータ処理部ブロック図を示す。このデータ処理部は、入力バッファ11、データ選択回路12、拡大縮小カウンタ13、出力バッファ14の他に、二値化回路30とディザデータ回路31とを備える。即ち、具体例1の回路に対し、新たに二値化回路30とディザデータ回路31とを追加した点が異なる。

【0044】ディザデータ回路31は、ディザデータ読み出し回路33と、ディザデータメモリ34とから成る。ディザデータメモリ34には、ディザデータ35が記憶されている。このディザデータ35は、例えば図2に示したCPU1によってディザデータメモリ34に書き込まれる。なお、ディザデータのデータ量が多い場合には、図の破線のブロックで示したように、ディザデータ自動転送のためのDMA回路32が利用される。

【0045】例えば入力バッファ11は、常に入力データの1画素分が8ビットの階調データを記憶する構成にしておく。従って、例えば実際にRAM4から読み出される入力データが1画素2ビット、あるいは1画素4ビットで表現されるものであっても、ここで1画素8ビットに正規化をしてしまう。こうして、入力バッファ11に保持された1画素8ビット1ライン分のデータが、二値化回路30により二値化処理される。

【0046】上記二値化回路30とディザデータ回路31の具体的な動作を説明する。二値化回路30には、データ選択回路12の出力する8ビットの階調データが1画素分ずつ順に送り込まれる。二値化回路30は、ディザデータ回路31から出力される閾値と、データ選択回

路12の出力とを比較し、二値化処理を行う。即ち、データ選択回路12の出力が閾値より大きい場合は“1”とし、それ以外の場合は“0”という出力を出力バッファ14に書き込む。

【0047】上記ディザデータは、例えばデータ選択回路12から出力される1ライン分の画素を処理するために必要な多数の閾値群から構成することができる。ディザデータ読み出し回路33は、ディザデータメモリ34から閾値群を順番に読み出し、先頭から最後まで読み出した後は再び先頭に戻って、該当する閾値を読み出すというように動作する。これによって、複雑に設計された多数の閾値を用いてデータ選択回路12の出力するデータの二値化を行うことができる。

【0048】1画素分のデータにそれぞれ1個の閾値を供給する場合には、例えば、ディザデータとして、1ライン分の閾値を用意して、各ラインの先頭から順に対応する閾値を供給して、次のラインに移るときは再び先頭から閾値を供給するように閾値の読み出し用カウンタをセットすればよい。また、例えば4×4マトリクスのディザパターンの場合には、4ライン分の閾値を用意する。ディザデータ中の各閾値を1ライン目から4ライン目まで順に読み出し、5ライン目の先頭で再びディザデータ中の先頭の閾値を読み出すというようにする。

【0049】例えば16×16マトリクスのディザパターンを採用するような場合には、ディザデータのデータ量が16ライン分となる。このような大量の閾値を全て格納しておくようなメモリをこの中に設けるのは不経済である。この場合には、ディザデータをRAM4に格納しておき、所定のタイミングで所定量ずつ分割して受け入れる図6の破線に示すようなダイレクトメモリアccess回路を設けるとよい。1画素分のデータに対して4個の閾値を順番に供給して、4個の二値化データを出力するような制御もできる。二値化処理以前のデータ転送クロックに対して二値化処理以後のデータ転送クロックを4倍にすればよい。

【0050】また、データ選択回路12から出力される各画素の階調データ1個に対し1個の閾値を対応させるように閾値の読み出しを行うと、例えば8ビットの階調データがそのまま二値化されて1ビットの印刷画像が得られる。即ち、入力バッファ11に保持された1ライン分の画像データが8分の1にデータ量を縮小されて、出力バッファ14に格納される。ライン長を2倍に拡大した場合でも、入力バッファの4分の1のデータが出力バッファに格納される。ライン長を2分の1にした場合には、入力バッファの16分の1のデータが出力バッファ14に格納される。

【0051】〈具体例2の効果〉以上のように、この具体例によれば、拡大縮小処理を終了し、二値化処理を終了して十分にデータ量が減少したものが、図1に示すRAM4に書き込まれる。しかも、そのデータはすぐに印

刷処理ができる印刷データになっている。こうしたことから、従来のように何度も入力データを読み出して演算処理を行い、最終的に印刷データを得るという場合に比べて、RAM4に対するデータ読み出し書き込み回数を大幅に削減できる。

【0052】〈具体例3〉上記のように、入力データを指定された倍率で拡大したり、縮小したり、二値化処理すると、出力データの一部が印刷可能領域を越えてしまうような場合がある。こうした場合、入力データの先頭部分や終端部分、あるいは出力データの先頭部分や終端部分の一部を除去する必要がある。上記のデータ処理部に、こうした不要データの削除機能を付加することによって、更にCPUの負担を軽減することが可能になる。この具体例3以降は、こうした機能を実現する。

【0053】図7は、具体例3のデータ処理部ブロック図である。この図の入力バッファ11の入力側に設けたオフセット制御回路40が、この具体例で新たに付け加えられたものである。それ以外の回路の構造や機能は、例えば具体例2のものと同じでよいので説明を省略する。

【0054】図7の回路の動作及びオフセット制御回路40の動作を説明するために、図7左下部分の一点鎖線に囲まれた場所にデータの内容を表示した。このデータ処理部は、図に示す例えば1ライン分の入力データD1を拡大し二値化処理して、1ライン分の出力データD0を得る機能を持つものとする。ここで、入力データD1の先頭部分に不要データDDが存在するとする。不要データとは、例えば、元データに描画関数による塗り潰し部がある場合で次の描画データが前の描画データと重なるようなときには、重なり部分は次の描画データにとっては不要なデータとなる。この不要データDDが例えば8画素分あるとした場合、オフセット制御回路40は、図1に示したDMA回路8から1画素分ずつ転送される入力データのはじめの8画素分のデータを遮断し、入力バッファ11への転送を阻止する。

【0055】こうした遮断回路は各種のものが考えられる。例えば1画素分のデータを一時保持するレジスタと、このレジスタに保持したデータを入力バッファ11へ転送する転送クロックを1ライン分のデータの先頭から8画素分だけマスクするゲート回路等により構成できる。またあるいは、入力バッファへ1画素分のデータを転送する回路中に挿入したゲート群と、そのゲートを1ライン分のデータの先頭から8画素分の転送時間だけ遮断するゲート制御回路により構成できる。

【0056】また、例えば512画素の入力データのうちの先頭8画素が不要データとしたとき、504画素分のデータが保持できるシフトレジスタを入力バッファ11として使用する。この入力バッファ11に512画素のデータを順に転送すると、最後の画素のデータを書き込んだとき、最初の8画素分のデータはオーバーフロー

して切り取られる。また、入力バッファ11を通常のメモリとする方法もある。この場合にははじめに入力バッファに全ての入力データが書き込まれる。データ選択回路12はCPUの指定したオフセット数を記憶しておく。そして、そのオフセット数だけ読み出し開始先頭アドレスをシフトさせる。

【0057】最初の2例は入力バッファ11の直前にオフセット制御回路40を配置する例である。また、残りの2つの例は、入力バッファ11とオフセット制御回路40とを一体に構成した例である。いずれの例も、従来から良く知られたデータ処理用の簡単なハードウェアを用いて実現できる。以下に説明する他の具体例においても、同様にこうした回路を利用することができる。

【0058】〈具体例3の効果〉以上のように、入力データの先頭部分の所定数のデータを削除する機能を持つオフセット制御回路40を設けることによって、出力データのライン長を調整し、そのまま拡大あるいは縮小処理して印刷制御部に転送できる印刷データを生成することが可能になる。これにより、CPUの負担を軽減できる。

【0059】〈具体例4〉上記の例では、入力データの先頭部分を削除する例を説明した。この具体例4の場合には、出力データの先頭の不要部分を削除する。図8には、具体例4のデータ処理部ブロック図を示す。この図に示すように、二値化回路30と出力バッファ14との間にオフセット制御回路41を設けている。このオフセット制御回路41以外の部分は、具体例2の回路と全く同様の構成でよい。従って、その他の部分の説明は省略する。

【0060】オフセット制御回路41は、図の左下側に示すように、二値化回路30から転送される出力データD0の先頭部分の所定長の不要データDDを削除する機能を持つ。オフセット制御回路41と出力バッファ14との関係は、具体例3のオフセット制御回路40と入力バッファ11の関係と同様である。故に、オフセット制御回路41は、具体例3と同様のハードウェア上により実現できる。

【0061】〈具体例4の効果〉以上のように、二値化後のデータの先頭部分を出力バッファ格納時に自動的に削除すれば、そのまま印刷データとして処理することが可能になる。

【0062】〈具体例5〉上記いずれの例も、例えば、1ページ分のイメージデータを処理する場合に、入力バッファに少なくとも1ライン分のデータを保持して処理すると効率よく処理ができる。しかしながら、入力バッファの容量や、その他様々な事情により、1ライン分のデータを一括して入力バッファに格納できないことがある。この場合には、1ラインのデータを数分の1ずつに分けて入力バッファに転送し、別々に拡大縮小処理することがある。この具体例では、このような場合の印刷デ

10

20

30

40

50

ータ長の調整を行う。

【0063】図9には、具体例5のデータ処理部ブロック図を示す。この図のライン長補正回路42を除く部分は、具体例4の回路と同様である。ライン長補正回路42は、出力バッファ14に書き込まれるデータのライン長を常に一定の長さに調整する機能を持つ。具体的には、図の右側に示すように、予め決められたライン長Lだけのデータを受け入れると、それ以上の余剰データを破棄する。

【0064】一方、出力バッファ14に書き込まれるデータのライン長が上記ライン長Lに満たない場合には、不足したビット数だけデータを補充する。具体的なデータ補充方法としては、書き込まれたデータの末尾のビットをコピーして、必要数だけ付加するといった方法による。

【0065】ライン長補正回路42は、例えばライン長Lに相当するデータを記憶するレジスタを備える。オフセット制御回路41には二値化処理後の所定長のデータが1画素分ずつ入力する。出力バッファ14は、オフセット制御回路41から入力したデータを1画素分ずつ受け入れて所定の領域に書き込む。このとき、ライン長補正回路42に記憶されたライン長分だけオフセット制御回路41の出力を取り込むようにする。従って、オフセット制御回路41からライン長補正回路42によって指定された長さ以上のデータが出力バッファ14に入力したとしても、出力バッファ14に余剰データは書き込まれない。

【0066】一方、オフセット制御回路41から出力されるデータ長がライン長補正回路42に記憶されたデータ長よりも短い場合、オフセット制御回路41は、最後の画素に相当するデータを保持したまま動作を停止する。出力バッファ14は、オフセット制御回路41からライン長分のデータを受け入れるために、オフセット制御回路41の保持したデータの読み取りを継続する。即ち、ライン長Lに達するまで同一のデータを繰り返し読み取る。こうして、不足データの補充が可能になる。

【0067】図9には、データ処理部の動作説明図を示す。上記の具体例の装置によれば、以下のような動作が実現する。(a)から(c)までが、その説明図である。図の(a)に示すようなライン長Lの1ライン分の出力データを得る処理を行う場合に、前述のように、入力データを分割して処理することを考える。例えば前半のL1と後半のL2とに分割して処理をすれば、両者をつなぎ合わせてライン長Lの1ライン分の出力データを得る。ところが、拡大縮小処理と二値化処理によって図の(b)に示すように、1回目の処理はL1にGだけデータが不足し、2回目の処理はL2にGだけデータが不足したとすると、出力データに隙間が生じ、画質が劣化する。

【0068】一方、前半も後半も、L1、L2の長さよ

りRだけ長さが増えると、1ライン分にまとめたとき、(c)に示すように、R分の重なりが生じてしまう。これも画質劣化の原因になる。図9に示したデータ処理部のライン長補正回路42は、自動的に出力バッファ14に格納するデータのライン長を調整する。なお、L1、L2はいずれも2/Lであればよいが、両者が異なってもよい。

【0069】〈具体例5の効果〉上記のように、ライン長補正回路42を用いてデータ長の調整を行いながら、出力バッファ14にデータの書き込みを行うので、拡大縮小処理や二値化処理によってデータ長が不揃いになった場合にも、一定の長さに揃えた出力データが得られる。これによって、後続回路の処理が簡単になる他、画質の劣化を防止することができる。

【0070】〈具体例6〉例えば、入力したイメージを100分の1とか1000分の1というように極端な縮小処理を行う場合、縮小処理後のデータ生成に必要なデータは元データのごく一部に過ぎなくなる。即ち、元データの1000個に1個の間隔で点在するデータを拾い上げながら処理を実行することになる。従って、RAM4に記憶された入力データを全てデータ処理部に転送し、その後縮小処理のための演算を行うのは記憶領域の無駄になる。そこで、この具体例では、図1に示したDMA回路8にスキップ機能を付加する。

【0071】図11は、具体例6の印刷データ生成装置ブロック図である。図の装置のDMA回路8には、スキップ制御回路43が接続されている。その他の回路は、具体例5のものと変わるところはない。なお、その他の回路については、具体例1以下のいずれの構成によっても構わない。ここで、再び図10を参照する。図10の(d)に示したものは、RAM4に記憶された入力データである。各画素のデータに順に“0”，“1”，“2”，“3”，…というように番号を付した。

【0072】例えばこのようなデータを20分の1に縮小することが予め分かっている場合には、上記DMA回路8は、RAM4中のデータを20画素分ずつスキップして読み出し、入力バッファ11に転送するように動作する。図には、読み出すデータを太い線の枠で囲んで例示した。これで入力バッファ11の記憶領域も十分縮小でき、DMA回路8によるRAM4へのアクセス回数も減少させることができる。

【0073】図2に示したCPU1は、図11に示した拡大縮小カウンタ13に、例えば20分1という縮小率の設定を行う。スキップ制御回路43は、縮小率を考慮したスタートアドレスと、スキップ量と、ストップアドレスをDMA回路8にセットする。これにより、DMA回路8は、入力バッファ11に対しその後の拡大縮小処理に必要なデータのみを転送する。なお、この具体例の動作はデータの縮小率が著しい場合に効果がある。従って、例えば縮小率が2分の1とか3分の1程度ではスキ

ップ制御回路43を動作させないほうが制御が簡単になることもある。この場合には、スキップ制御回路43に対しCPU1がスキップ制御を有効にするかしないかの情報をセットするようにしてもよい。スキップ機能を有効したほうがよいが無効にしたほうがよいかといった判断の基準は、CPU1のプログラム中に書き込んでおけばよい。

【0074】なお、具体例1等で説明したデータ処理回路では、拡大縮小カウンタ13が入力バッファ11に記憶されたデータのアドレスを計算している。データ選択回路12はこのアドレスを使用して入力バッファ11中のデータの選択を行う。従って、具体例1等で説明したのと同様の動作を行うためには、入力バッファ11中にRAM4に記憶された入力データ全てがそのまのアドレスで書き込まれていることが好ましい。

【0075】この場合には、DMA回路8が入力バッファ11に“0”番目のデータを転送した後、“20”番目のデータを転送するまでの間に、入力バッファ11に、2番目～19番目の分のダミーデータを書き込むようにすればよい。これには、DMA回路8の20倍の速度の制御クロックで入力バッファ11の書き込み動作をさせればよい。ダミーデータの内容は任意である。この場合でも、DMA回路8によるRAMのアクセス回数を減少させて、バス占有時間を減らす効果がある。

【0076】もちろん、入力バッファ11にDMA回路8がスキップ読み出しをしたデータをそのまま順番に書き込むこともできる。この場合には、既に入力データの縮小処理が終了しているから拡大縮小カウンタ13は作動しない。データ選択回路12は入力バッファ11に書き込まれたデータをそのまま二値化回路30に出力すれ

*ばよい。

【0077】〈具体例6の効果〉上記のように、DMA回路8がスキップ読み出しによりRAM4から入力バッファ11に対しデータの転送を行うので、全てのデータをRAM4から入力バッファ11に転送する場合に比べて、DMA回路8によるデータ読み出し処理時間を短縮することが可能になる。また、これによって、CPUのバスを占有する時間を短縮することが可能になる。

【図面の簡単な説明】

【図1】具体例1の印刷データ生成装置ブロック図である。

【図2】印刷装置のブロック図である。

【図3】データ処理部の機能説明図である。

【図4】拡大縮小カウンタのブロック図である。

【図5】拡大縮小動作説明図である。

【図6】具体例2のデータ処理部ブロック図である。

【図7】具体例3のデータ処理部ブロック図である。

【図8】具体例4のデータ処理部ブロック図である。

【図9】具体例5のデータ処理部ブロック図である。

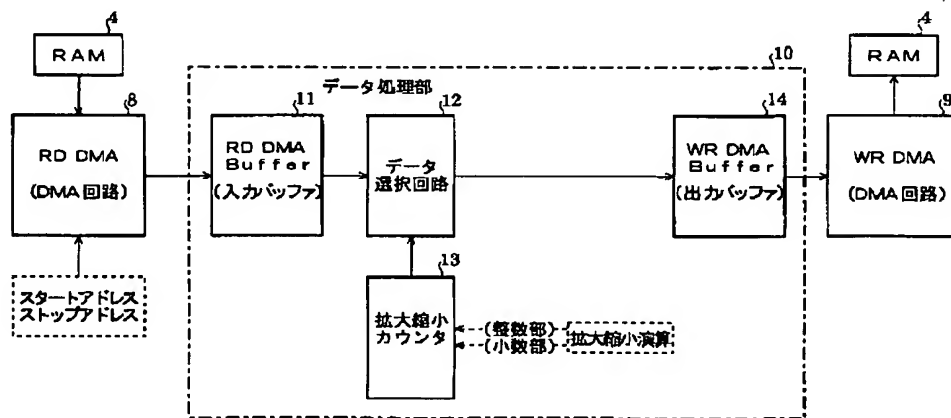
【図10】データ処理部の動作説明図である。

【図11】具体例6の印刷データ生成装置ブロック図である。

【符号の説明】

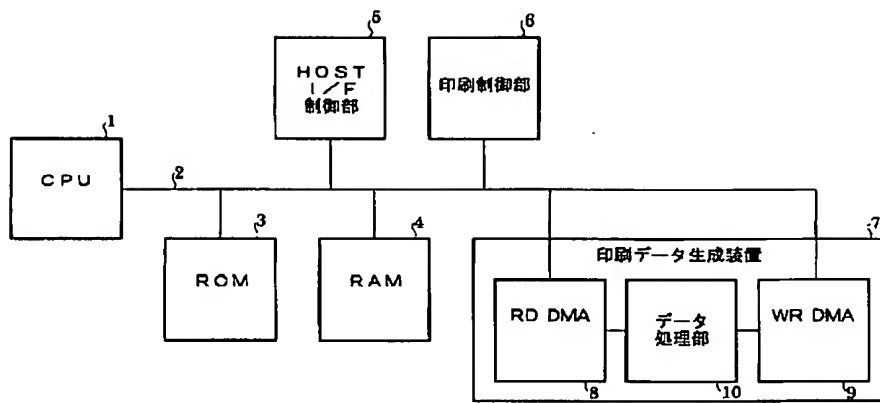
- 4 RAM
- 8, 9 DMA回路
- 10 データ処理部
- 11 入力バッファ
- 12 データ選択回路
- 13 拡大縮小カウンタ
- 14 出力バッファ

【図1】



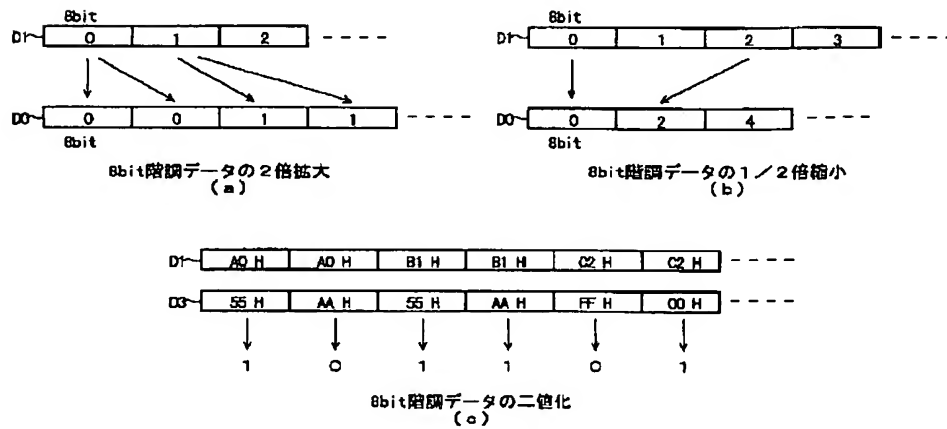
具体例1の印刷データ生成装置ブロック図

【図2】



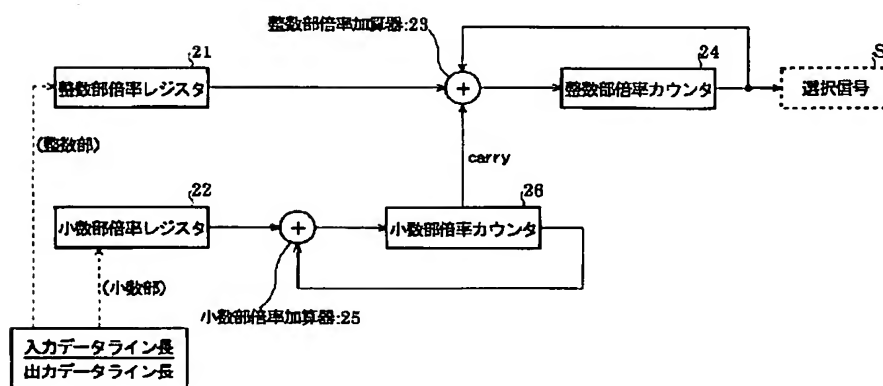
印刷装置のブロック図

【図3】



データ処理部の機能説明図

【図4】



拡大縮小カウンタのブロック図

【図5】

出力データ 要素位置	1/2倍縮小	1/3倍縮小	1/4倍縮小	1/5倍縮小
	整数部=2 小数部=0.0	整数部=3 小数部=0.0	整数部=4 小数部=0.0	整数部=5 小数部=0.0
	Rreg: Preg	Rreg: Preg	Rreg: Preg	Rreg: Preg
0	0.0 0	0.0 0	0.0 0	0.0 0
1	0.0 2	0.0 3	0.0 4	0.0 5
2	0.0 4	0.0 6	0.0 8	0.0 10
3	0.0 6	0.0 9	0.0 12	0.0 15
4	0.0 8	0.0 12	0.0 16	0.0 20
5	0.0 10	0.0 15	0.0 20	0.0 25

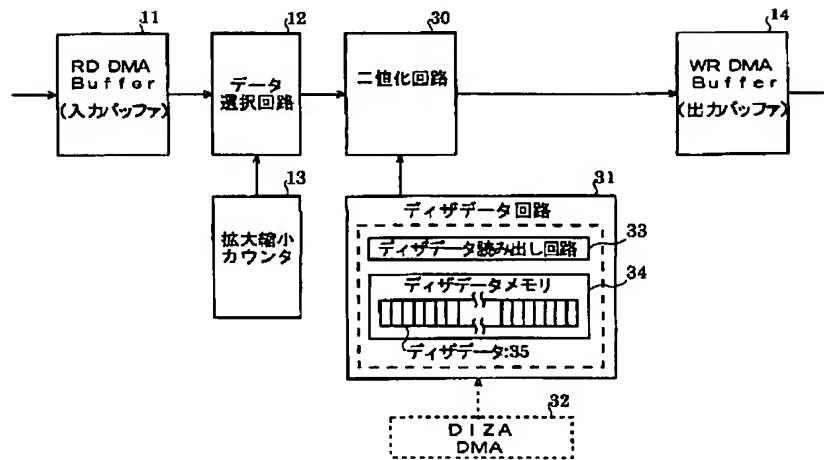
(a)

出力データ 要素位置	2倍拡大	3/2倍拡大	5/2倍拡大
	整数部=0 小数部=0.60	整数部=0 小数部=0.66	整数部=0 小数部=0.40
	Rreg: Preg	Rreg: Preg	Rreg: Preg
0	0.00 0	0.00 0	0.00 0
1	0.50 0	0.66 0	0.40 0
2	1.00(-0.00) 1	1.32(-0.32) 1	0.80 0
3	0.50 1	0.66 1	1.20(-0.20) 1
4	1.00(-0.00) 2	1.64(-0.64) 2	0.80 1
5	0.60 2	1.30(-0.30) 3	1.00(-0.00) 2
6	1.00(-0.00) 3	0.98 3	0.40 2
7	0.50 3	1.62(-0.62) 4	0.80 2
8	1.00(-0.00) 4	1.28(-0.28) 5	1.20(-0.20) 3
9	0.50 4	0.64 5	0.60 3
10	1.00(-0.00) 5	1.60(-0.60) 6	1.00(-0.00) 4

(b)

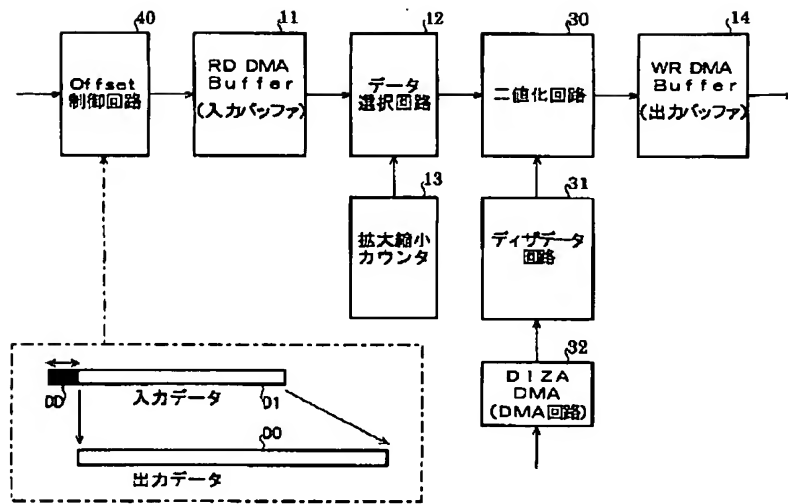
拡大縮小動作説明図

【図6】



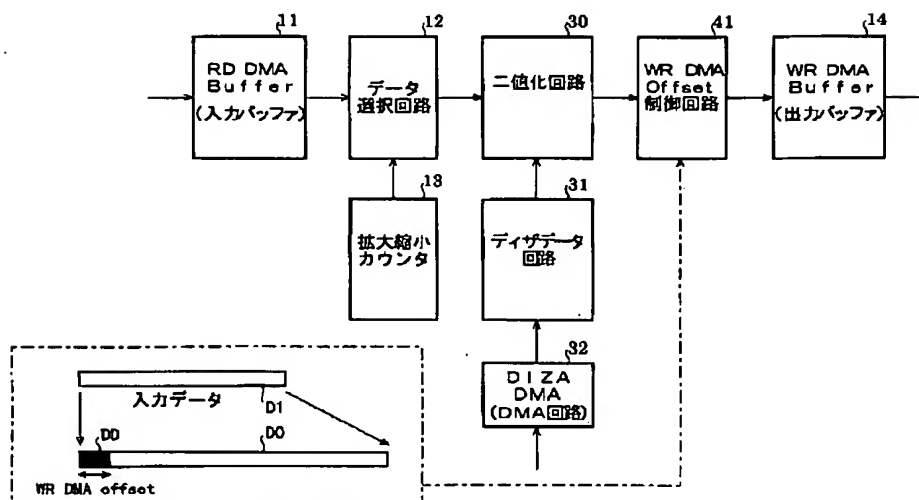
具体例2のデータ処理部ブロック図

【図7】



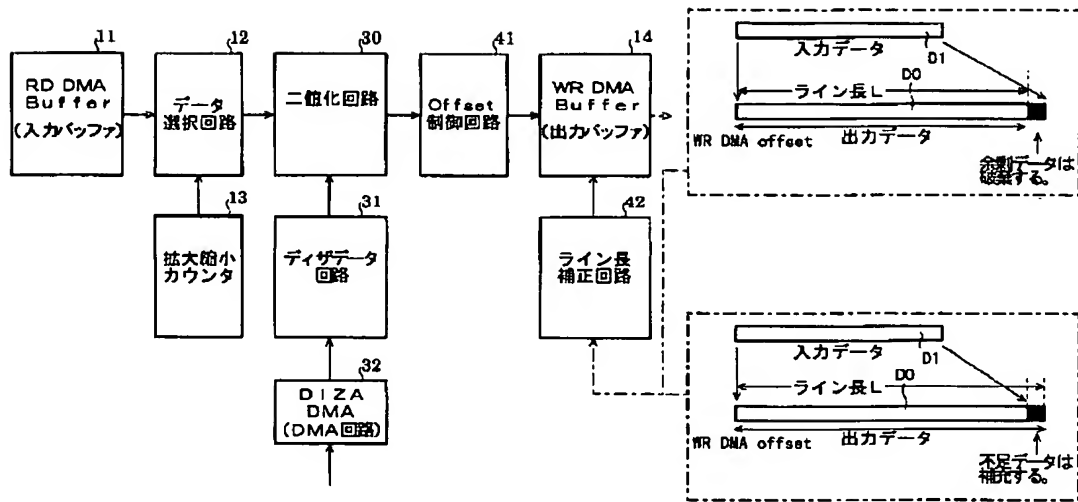
具体例3のデータ処理部ブロック図

【図8】



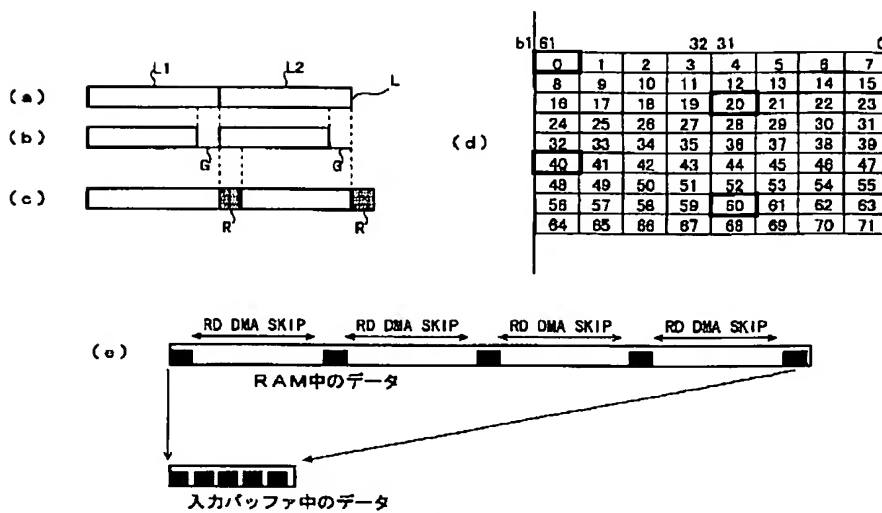
具体例4のデータ処理部ブロック図

【図9】



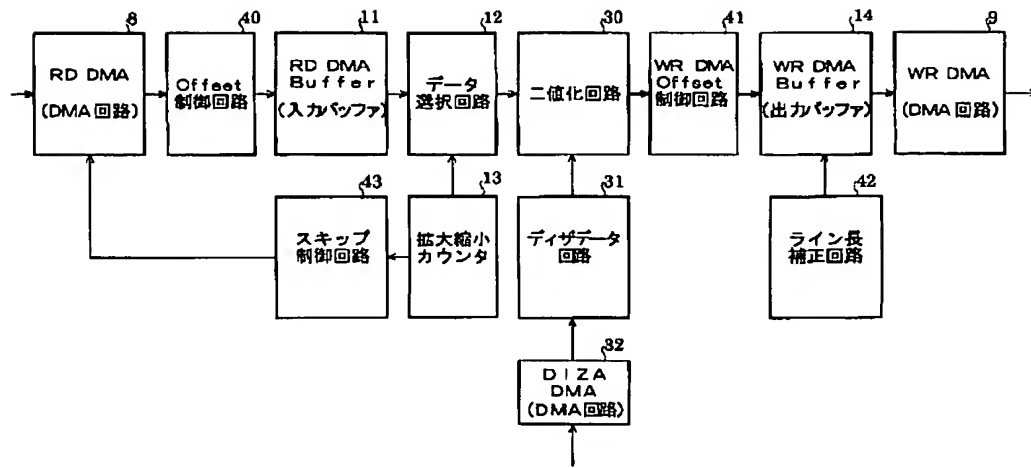
具体例6のデータ処理部ブロック図

【図10】



データ処理部の動作説明図

【図11】



具体例6の印刷データ生成装置ブロック図

フロントページの続き

(72)発明者 遠藤 義和
東京都港区芝浦四丁目11番地22号 株式会
社沖データ内

Fターム(参考) 2C087 BA03 BC02 BC05 BD06 BD24
CA13
5B021 AA01 AA02 LB07 LG08
9A001 HH23 HH24 HH27 JJ35 KK42

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-150739

(43)Date of publication of application : 05.06.2001

(51)Int.Cl.

B41J 5/30
G06F 3/12

(21)Application number : 11-337158

(71)Applicant : OKI DATA CORP

(22)Date of filing : 29.11.1999

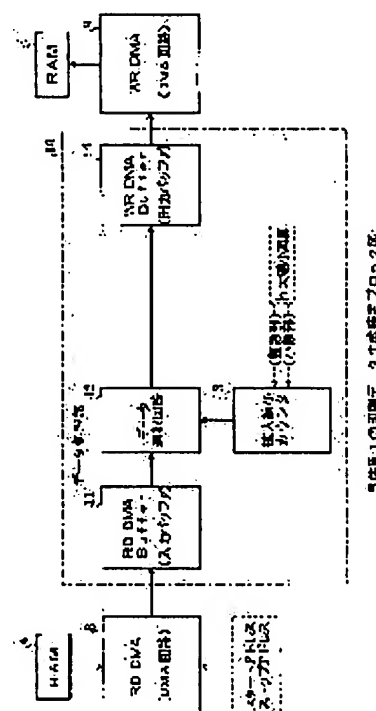
(72)Inventor : YAMAMOTO SATOSHI
ISHIKAWA OSAMU
ENDO YOSHIKAZU

(54) PRINT DATA GENERATOR

(57)Abstract:

PROBLEM TO BE SOLVED: To increase the speed of print data generation processing without imposing a load on a CPU by constituting a magnifying/contracting section and a binarizing section out of hardware.

SOLUTION: An input buffer 11 holds input data. A magnifying/contracting counter 13 receives the integer part and decimal part of results obtained by dividing the line length of input data by the line length of output data and generates a signal for selecting any data held in the input buffer 11. A data select circuit 12 selects data held in the input buffer 11 and delivers it to an output buffer 14. A data processing section 10 is constituted of hardware.



LEGAL STATUS

[Date of request for examination] 20.01.2003

[Date of sending the examiner's decision of rejection] 05.12.2006

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The input buffer holding input data, and the integer part and fraction part of a result which did the division of the Rhine length of said input data in the Rhine length of output data are accepted. The enlarging-or-contracting counter which generates and outputs the selection signal which chooses the data for 1 pixel held at said input buffer, Print-data generation equipment characterized by having chosen the data for 1 pixel held at said input buffer by the selection signal which said enlarging-or-contracting counter outputs in predetermined order, and having the data selection circuitry which generates the output data which carried out enlarging or contracting of said input data.

[Claim 2] In print-data generation equipment according to claim 1 an enlarging-or-contracting counter The integer part scale-factor register holding the integer part of the result of having done the division of the Rhine length of input data in the Rhine length of output data, The fraction part scale-factor register holding the fraction part of the result of having done the division of the Rhine length of input data in the Rhine length of output data, The address which reads the data for 1 pixel arranged at the head of input data is made into initial value. The integer part scale-factor counter which carries out accumulation of the integer part which said integer part scale-factor register holds, and outputs it for every data read-out for 1 pixel, When accumulation of the fraction part which said fraction part scale-factor register holds is carried out for every data read-out for 1 pixel by having made zero into initial value and this accumulation value exceeds 1, Print-data generation equipment characterized by having the fraction part scale-factor counter which outputs the data for adding 1 to said integer part.

[Claim 3] The input buffer holding the input data containing gradation data, and the integer part and fraction part of a result which did the division of the Rhine length of said input data in the Rhine length of output data are accepted. The enlarging-or-contracting counter which generates and outputs the selection signal which chooses the data for 1 pixel held at said input buffer, The data for 1 pixel held at said input buffer by the selection signal which said enlarging-or-contracting counter outputs are chosen in predetermined order. The data selection circuitry which generates the output data which carried out enlarging or contracting of said input data, Print-data generation equipment characterized by having the dither data circuit which reads and outputs the threshold which constitutes dither data in predetermined order, and the binarization circuit which accepts said threshold and carries out binarization of the output of said selection circuitry for every data read-out for 1 pixel.

[Claim 4] It is print-data generation equipment characterized by having the Direct-Memory-Access circuit in which a dither data circuit divides external memory to dither data into in print-data generation equipment according to claim 3, and it accepts.

[Claim 5] Print-data generation equipment characterized by being contained in the input side of an input buffer at the head of input data, having held the number of invalid datas with unnecessary including in output data in print-data generation equipment according to claim 1, and preparing the offset control circuit which controls the write-in starting position to the input buffer of input data to shift only said invalid number of data.

[Claim 6] Print-data generation equipment characterized by having held the number of invalid datas with unnecessary including in output data at the input side of an output buffer, and preparing the offset control circuit which controls the write-in starting position to the output buffer of output data to shift only said invalid number of data in print-data generation equipment according to claim 1.

[Claim 7] The output buffer which holds output data in print-data generation equipment according to claim 1, When the output-data length which holds the convention data length set up beforehand and is written in said output buffer exceeds said convention data length When the output-data length which cancels the part beyond said convention data length of output data, and is written in said output buffer does not fulfill said convention data length Print-data generation equipment characterized by having the Rhine length amendment circuit which controls the writing of the output data to said output buffer to fill up the part with which said convention data length of output data is not filled.

[Claim 8] Print-data generation equipment characterized by preparing the skip control circuit controlled so that said DMA circuit skips and reads input data from said external memory at spacing according to reduction percentage in print-data generation equipment according to claim 1, when generating the output data which reduced the input data, the DMA circuit which transmits input data to an input buffer from external memory.

[Translation done.]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the print-data generation equipment which performs enlarging or contracting of an image data with gradation nature, and binarization processing.

[0002]

[Description of the Prior Art] When printing the image data generated by computer by the printer, transform processing of the data according to the function of a printer is performed beforehand. For example, in order to double the resolution of the inputted image data for printing with the print resolution of a printer, expansion or contraction processing of an image data is performed. In expansion processing, the image data of every one line is memorized to buffer memory, the data of each pixel are copied by magnifying power, and the number of pixels is increased. A copy increases the number of Rhine similarly. On the other hand, in contraction processing, the image data for one line is stored in buffer memory, infanticide processing of a pixel is performed, the number of pixels is reduced to it, and the result is outputted to it. The number of Rhine for which it applies is also reduced.

[0003] Moreover, many printers express an image by binary control whether a black dot is printed or it carries out. Therefore, in the case of the gradation data in which each pixel has the information showing a shade, binarization processing which used the dither matrix is performed. In such processing, every one gradation data of each pixel is read, it is compared with the threshold which constitutes a dither matrix, and binarization is performed.

[0004]

[Problem(s) to be Solved by the Invention] By the way, there were the following technical problems which should be solved in the above Prior arts. If print data are received from high order equipments, such as a computer, a printer will perform the above enlarging-or-contracting processings and binarization processing, and will generate print data. CPU (central processing unit) of a printer calls print data to buffer memory per the Rhine unit or pixel, performs data processing for enlarging or contracting or binarization, and turns and outputs the result to a print engine.

[0005] However, many repeats of data processing to which processing of this kind made the data for 1 pixel the unit are contained, and when it is the image data of high resolution, that amount of data processing becomes huge. And 1 pixel of the amount of gradation data is about 8-bit data, and even if it is CPU in which bus width of face has 32 bits or a size beyond it, efficient data processing which used the bus width of face effectively is difficult. That is, the high-speed data-processing capacity of CPU original for bus width of face to be wide could not be demonstrated effectively, but there was a problem of taking data-processing time amount for a long time for the frequent middle data access to buffer memory. Moreover, when a big load was applied to such print-data generation processing, there was also a problem of pressing the control function of other parts of the printer by CPU.

[0006]

[Means for Solving the Problem] This invention adopts the next configuration in order to solve the above point.

<Configuration 1> The input buffer holding input data, and the integer part and fraction part of a result which did the division of the Rhine length of the above-mentioned input data in the Rhine length of output

data are accepted. The enlarging-or-contracting counter which generates and outputs the selection signal which chooses the data for 1 pixel held at the above-mentioned input buffer, Print-data generation equipment characterized by having chosen the data for 1 pixel held by the selection signal which the above-mentioned enlarging-or-contracting counter outputs at the above-mentioned input buffer in predetermined order, and having the data selection circuitry which generates the output data which carried out enlarging or contracting of the above-mentioned input data.

[0007] <Configuration 2> In print-data generation equipment given in a configuration 1 an enlarging-or-contracting counter The integer part scale-factor register holding the integer part of the result of having done the division of the Rhine length of input data in the Rhine length of output data, The fraction part scale-factor register holding the fraction part of the result of having done the division of the Rhine length of input data in the Rhine length of output data, The address which reads the data for 1 pixel arranged at the head of input data is made into initial value. The integer part scale-factor counter which carries out accumulation of the integer part which the above-mentioned integer part scale-factor register holds, and outputs it for every data read-out for 1 pixel, When accumulation of the fraction part which the above-mentioned fraction part scale-factor register holds is carried out for every data read-out for 1 pixel by having made zero into initial value and this accumulation value exceeds 1, Print-data generation equipment characterized by having the fraction part scale-factor counter which outputs the data for adding 1 to the above-mentioned integer part.

[0008] <Configuration 3> The input buffer holding the input data containing gradation data, The enlarging-or-contracting counter which generates and outputs the selection signal which chooses the data for 1 pixel which accepted the integer part and fraction part of a result which did the division of the Rhine length of the above-mentioned input data in the Rhine length of output data, and was held at the above-mentioned input buffer, The data for 1 pixel held by the selection signal which the above-mentioned enlarging-or-contracting counter outputs at the above-mentioned input buffer are chosen in predetermined order. The data selection circuitry which generates the output data which carried out enlarging or contracting of the above-mentioned input data, Print-data generation equipment characterized by having the dither data circuit which reads and outputs the threshold which constitutes dither data in predetermined order, and the binarization circuit which accepts the above-mentioned threshold and carries out binarization of the output of the above-mentioned selection circuitry for every data read-out for 1 pixel.

[0009] <Configuration 4> It is print-data generation equipment characterized by having the Direct-Memory-Access circuit which a dither data circuit divides external memory to dither data in print-data generation equipment given in a configuration 3, and accepts.

[0010] <a configuration 5> -- the number of invalid datas with unnecessary it being contained in the configuration 1 at the head of input data in the print-data generation equipment of a publication at the input side of an input buffer, and including in output data -- holding -- the write-in starting position to the input buffer of input data -- the above -- the print-data generation equipment characterized by preparing the offset control circuit controlled to shift only the invalid number of data.

[0011] <a configuration 6> -- the number of invalid datas with unnecessary including in the input side of an output buffer in print-data generation equipment given in a configuration 1 at output data -- holding -- the write-in starting position to the output buffer of output data -- the above -- the print-data generation equipment characterized by preparing the offset control circuit controlled to shift only the invalid number of data.

[0012] <Configuration 7> The output buffer which holds output data in the print-data generation equipment of a publication in a configuration 1, When the output-data length which holds the convention data length set up beforehand and is written in the above-mentioned output buffer exceeds the above-mentioned convention data length When the output-data length which cancels the part beyond the above-mentioned convention data length of output data, and is written in the above-mentioned output buffer does not fulfill the above-mentioned convention data length Print-data generation equipment characterized by having the Rhine length amendment circuit which controls the writing of the output data to the above-mentioned output buffer to fill up the part with which the above-mentioned convention data length of output data is not filled.

[0013] <Configuration 8> Print-data generation equipment characterized by to prepare the skip control circuit controlled so that the above-mentioned DMA circuit skips and reads input data from the above-mentioned external memory at spacing according to reduction percentage in print-data generation equipment

given in a configuration 1, when generating the output data which reduced the input data, the DMA circuit which transmits input data to an input buffer from external memory.

[0014]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained using an example.

<Example 1> In this invention, in order to mitigate the burden of CPU, data processing of a pixel unit is performed as much as possible using hardware. Drawing 1 is the block diagram showing the print-data generation equipment of an example 1. The equipment of drawing is constituted so that enlarging-or-contracting processing may be carried out in the data-processing section 10 and the input data read from RAM (random access memory)4 using the DMA circuit (RD DMA (Direct-Memory-Access controller)) 8 may be written in RAM4 using the DMA circuit 9. The data-processing section 10 is constituted by an input buffer 11, the data selection circuitry 12, the enlarging-or-contracting counter 13, and the output buffer 14.

[0015] Before explaining concrete actuation of this circuit, the configuration of the whole equipment is explained first. Drawing 2 is the block diagram showing the whole airline printer configuration. The equipment of drawing is constituted so that CPU1 may control each part of equipment through a bus line 2. ROM (read-only memory)3, RAM4, the host interface control section 5, the printing control section 6, and print-data generation equipment 7 are connected to the bus line 2. The DMA circuit 8 for read-out (RD DMA), the data-processing section 10, and the DMA circuit 9 for writing (WR DMA) are established in print-data generation equipment 7.

[0016] ROM3 is the memory which stored the program for actuation of CPU1 etc. RAM4 is the memory for storing the input data or print data used as the operational parameter and processing object by CPU1. The host interface control section 5 is the part which performs communications control of the high order equipment and the printer which are not illustrated, and input data inputs it from high order equipment through this host interface control section 5, and it is memorized by RAM4. The printing control section 6 is a part which controls the print engine which is not illustrated and controls printing of the print data memorized by RAM4.

[0017] Print-data generation equipment 7 is a part which performs the enlarging-or-contracting processing and binarization processing of input data which were accepted in RAM4 through the host interface control section 5, and generates print data. The DMA circuit 8 for read-out is a circuit which reads input data from RAM4 in order, and is supplied to the data-processing section 10. The DMA circuit 9 for writing is a circuit which performs actuation which writes in order the print data which processing ended in the data-processing section 10 in RAM4.

[0018] In this invention, the data-processing section 10 of this print-data generation equipment 7 is constituted by hardware, and performs, without depending enlarging-or-contracting processing and binarization processing on data processing of CPU1. It writes in with the DMA circuit 8 for read-out, and the DMA circuit 9 of business is all the Direct-Memory-Access controller generally known conventionally, and it is constituted by hardware.

[0019] The functional description Fig. of the data-processing section is shown in drawing 3. (a) illustrates the contents of the expansion processing facility. (b) illustrates the contents of the contraction processing facility. (c) illustrates the contents of the binarization processing facility. In expanding the input data D1 which expresses 1 pixel with 8-bit digital data twice as shown in (a) of drawing for example, the input data for 1 pixel is copied to output data D0 by a unit of 2 times, and it outputs it to them.

[0020] In the contraction processing shown in (b), input data D1 is taken out alternately per every pixel, and it considers as output data D0. That is, it reduces by infanticide processing. By binarization processing shown in (c), the threshold included in the dither pattern data D3 which are explained later is read in predetermined sequence, and it compares with the pixel value of each pixel of input data D1. In this way, let the data by which binarization was carried out be print data.

[0021] It returns to drawing 1 again and print-data generation equipment including the data-processing section 10 shown in drawing 1 is explained. The DMA circuit 8 for read-out is a circuit which performs processing which reads automatically the input data which accepts the start address and the stop address of input data from CPU1 shown in drawing 2, and corresponds from RAM4 and which was known well. This input data is written in constant-rate [every] sequence at an input buffer 11. Although it is desirable that the

data for at least one line can be written in, every 1/2 line or every several lines are sufficient. Termination of enlarging-or-contracting processing of the input data held at the input buffer 11 sends input data into an input buffer 11 in order with the procedure of requiring a transfer of the following input data from the DMA circuit 8.

[0022] The data selection circuitry 12 is a circuit which accepts the selection signal outputted from the enlarging-or-contracting counter 13, reads the data for 1 pixel with which it corresponds of the input data memorized by the input buffer 11, and is performed by repeating the processing which transmits this to an output buffer 14. For example, when an input buffer 11 considers as the memory which can hold the gradation data for one line, the data selection circuitry 12 supplies the selection signal outputted from the enlarging-or-contracting counter 13 as an address signal of an input buffer 11, and constitutes it by the bus line and gate circuit which transmit the data for 1 pixel read from the input buffer 11 to an output buffer 14.

[0023] An output buffer 14 consists of memory memorized in the order which received the data received through the data selection circuitry 12. Although a certain thing of the memory space is above desirable by 1 Rhine of an output image, it does not interfere by the capacity of arbitration like an input buffer 11. The DMA circuit 9 for writing is a circuit which performs control which transmits the print data which processing of the input data held at the input buffer 11 was completed, operated to the timing by which new input data is written in an input buffer 11, and were memorized by the output buffer 14 to RAM4. In this way, the transmitted print data are written in the predetermined field of RAM4 in order, and turn into print data for 1 page as a whole.

[0024] The block diagram of an enlarging-or-contracting counter is shown in drawing 4. The above-mentioned enlarging-or-contracting counter 13 is constituted by the integer part scale-factor register 21, the fraction part scale-factor register 22, the integer part scale-factor adder 23, the integer part scale-factor counter 24, the fraction part scale-factor adder 25, and the fraction part scale-factor counter 26 as shown, for example in this drawing.

[0025] When performing enlarging-or-contracting processing, CPU1 shown in drawing 2 searches for the result of having done the division of the Rhine length of input data in the Rhine length of output data, beforehand. It is "2" if it is twice the magnifying power of this, and it is "0.5" and 1/2. And it writes in the integer part scale-factor register 21, the integer part, "0". [for example,] Moreover, fraction part, "5", is written in the fraction part scale-factor register 22. [for example,] After making preparations including such processing, actuation of the above-mentioned data-processing section 10 is started.

[0026] The integer part scale-factor adder 23 adds the contents memorized by the integer part scale-factor register 21, the carry outputted from the fraction part scale-factor counter 26, and the output of the integer part scale-factor counter 24, and has the function which the integer part scale-factor counter 24 is made to memorize to the following timing. The fraction part scale-factor adder 25 adds the output of the fraction part scale-factor register 22, and the output of the fraction part scale-factor counter 26, and has the function which the fraction part scale-factor counter 26 is made to memorize to the following timing.

[0027] The integer part scale-factor counter 24 and the fraction part scale-factor counter 26 of initial value are "0." And timing is doubled with the control clock for processing every 1 pixel of input data, and this data processing is performed. That is, the output in the case of processing the very first pixel is "0" of the initial value memorized by the integer part scale-factor counter 24. And if the following control clock inputs, the addition result of the integer part scale-factor adder 23 will be memorized by the integer part scale-factor counter 24. And this serves as the next output. In addition, the selection signal S which this integer part scale-factor counter 24 outputs turns into an address signal for reading the specific pixel in the input data for one line memorized by the input buffer 11, for example.

[0028] Drawing 5 is the explanatory view of enlarging-or-contracting actuation. The principle of operation of the above-mentioned enlarging-or-contracting counter and the contents of the concrete output are explained using this drawing. Drawing 5 shows the relation between the pixel location of the output data in contraction processing, the contents (integer part =) of the integer part scale-factor register 21, the contents (fraction part =) of the fraction part scale-factor register 22, the contents Preg of the fraction part scale-factor counter 26, and the contents Rreg of the integer part scale-factor counter 24. (a) shows contraction processing and (b) shows expansion processing.

[0029] Here, the case where 1 contraction processing for 2 minutes is performed is considered. At this time,

the value which $\frac{\text{input data line length}}{\text{output data line length}}$ is set to "2." "2" of this division result is written in the integer part scale-factor register 21. Fraction part, "0.0", is written in the fraction part scale-factor register 22. [i.e.,] The contents Rreg of the fraction part scale-factor counter 26 are initial value "0.0." The contents Preg of the integer part scale-factor counter 24 are initial value "0."

[0030] First, to the timing which processes the very first pixel, since initial value "0" is stored in the integer part scale-factor counter 24, the value is outputted as a selection signal S. If the input data for one line is held at the input buffer 11, the data whose address is the pixel of "0, i.e., a head," will be written in as a pixel of the head of an output buffer 14. An input of the following control clock adds "2" memorized by the integer part scale-factor register 21 and "0" memorized by the integer part scale-factor counter 24.

[0031] Initial value "0" is memorized by the fraction part scale-factor counter 26, and "0.0" is written in the fraction part scale-factor register 22. Therefore, the addition result of the fraction part scale-factor adder 25 is "0", and this is again written in the fraction part scale-factor counter 26. For this reason, the fraction part scale-factor counter 26 does not generate a carry. Consequently, the addition result 2 of "0" memorized by "2" memorized by the integer part scale-factor register 21 and the integer part scale-factor counter 24 is written in the integer part scale-factor counter 24. In this way, the selection signal S "2" is outputted to the following timing.

[0032] Thus, since the selection signal was set to "2", the pixel of "2" is read for the address of input data, and the degree of "0" is written in an output buffer 14. Similarly, if the following control clock inputs, since the contents of the integer part scale-factor counter 24 are added with "2" of the contents of the integer part scale-factor register 21 by "2", a selection signal S will be set to "4." That is, as indicated in the train of Preg displayed as 1/2 time contraction of drawing 5 (a), a selection signal S is outputted like "0", "2", "4", "6", "8", and --. Consequently, the pixel in input data is chosen alternately and memorized by the output buffer 14. The above-mentioned enlarging-or-contracting counter 13 has the function to choose either of the input data which did in this way and was memorized by the input buffer 11, and to transmit the result of having carried out enlarging-or-contracting processing to an output buffer 14.

[0033] Next, expansion processing is explained. For example, in expanding input data twice, the Rhine length of input data is $\frac{\text{input data}}{2}$ in the Rhine length of output data, and it obtains the result "0.5." The data "0" are written in the integer part scale-factor register 21. On the other hand, the result "0.5" is written in the fraction part scale-factor register 22. In addition, precision of the fraction part scale-factor register 22 is made into double figures below decimal point, and the numeric value "0.50" is memorized to the fraction part scale-factor register 22 in the example of drawing 5 (b). What is necessary is to process a fraction by counting fractions as one, a cut-off, etc., and just to ask for fraction part, if a fraction is in a division result.

[0034] Here, when the first control clock inputs, "0" which is the initial value memorized by the integer part scale-factor counter 24 is outputted as a selection signal S like the time of 1/2 time contraction. Therefore, it does not change that the data of the first pixel are outputted as a pixel of the head of an output buffer 14. If the following control clock inputs, the aggregate value of the output of the integer part scale-factor register 21 and the value memorized by the integer part scale-factor counter 24 until now will input into the integer part scale-factor counter 24. In this case, since both are "0", "0" is memorized as it is by the integer part scale-factor counter 24.

[0035] The value of "0.5" memorized by the fraction part scale-factor register is memorized by the fraction part scale-factor counter 26. The fraction part scale-factor counter 26 will output a carry, if the data memorized here amount to "1." Since a carry is not outputted for the contents of the fraction part scale-factor counter 26 in the case of "0" and "0.5", from the integer part scale-factor counter 24, the selection signal S of "0" is outputted for the contents this time also.

[0036] Furthermore, the case where the following control clock inputs is considered. At this time, "0.5" is memorized by the fraction part scale-factor counter 26, and the fraction part scale-factor adder 25 adds the output of the fraction part scale-factor register 22, and the output of the fraction part scale-factor counter 26, and obtains the result "1.00." If this is sent into the fraction part scale-factor counter 26, for an overflow, the fraction part scale-factor counter 26 will store "00", and will output a carry to the integer part scale-factor adder 23. The integer part scale-factor counter 24 has memorized "0" until now. The contents of the integer part scale-factor register 21 are "0." The integer part scale-factor adder 23 outputs "1" which it is as a result of [these] addition to the integer part scale-factor counter 24. In this way, as for the integer part scale-factor

counter 24, the contents output the selection signal S of "1." Consequently, as shown in drawing 5 (b), the selection signal "1" is outputted to the 3rd time.

[0037] If the same processing is repeated, a selection signal will change like "0", "0", "1", "1", "2", "2", "3", "3", and --. Consequently, each pixel of input data overlaps by a unit of 2 times, it is transmitted to an output buffer 14, and the input data for one line is expanded twice in the direction of Rhine. Thus, the fraction part scale-factor counter 26 will output a carry, if accumulation of the fraction with which "1" is not filled is carried out and the accumulation result becomes more than "1", and it carries out the operation which adds "1" to the integer part scale-factor counter 24. When the magnifying power of arbitration is specified, the pixel of a required number is copied to predetermined timing by this, and the print data of the corresponding magnifying power can be obtained by it. Although concrete explanation was omitted in order to process other scale factors in the same procedure, the example was illustrated to drawing 5 (a) and (b).

[0038] In addition, binary data or gradation data is sufficient as input data in the above-mentioned example. At this time, it is good to arrange the data for 1 pixel with 8 bits irrespective of the number of bits of input data. It is for processing input data uniformly by single hardware. In this case, if the input data for 1 pixel is 8 bits or less in number of bits, the DMA circuit 8 should just write input data in the storage region of the 8-bit width of face of an input buffer 11 in order as it is. On the other hand, if the input data for 1 pixel may exceed 8 bits, the circuit which normalizes input data uniformly will be prepared. Namely, CPU prepares the register which writes in the number of bits of input data beforehand, calculates the product of the value and input data which did the division of the number of bits "8" of output data with this register value, and should just add the multiplication circuit which normalizes data automatically. For example, in order for input data to normalize 4 bits [per pixel] gradation data to 8 bits, processing which doubles data 17 for every pixel is performed.

[0039] Moreover, although the above-mentioned example showed only enlarging-or-contracting processing of the direction of Rhine of data (main scanning direction), enlarging-or-contracting processing of the direction of vertical scanning can be performed similarly. However, since contraction processing should just operate each Rhine on a curtailed schedule, it is [that expansion processing of the direction of vertical scanning should just copy each Rhine by magnifying power] easy data processing compared with processing of a main scanning direction. Moreover, since a data length is also long, it is suitable for high-speed data processing by CPU. Therefore, it is desirable that accept the output of the above-mentioned circuit and CPU carries out software processing. Of course, enlarging-or-contracting processing of the direction of vertical scanning may be beforehand substituted for the input side of the above-mentioned circuit. Thereby, expansion of a hardware scale can be prevented compared with the case where the direction of a main scanning direction and vertical scanning is also processed by hardware, and a miniaturization and low-cost-izing of equipment can be attained. In addition, enlarging-or-contracting processing of an image data may be performed per page, and you may carry out per object of the size of arbitration.

[0040] <Effectiveness of an example 1> As mentioned above, above print-data generation equipment accepts the integer part and fraction part of a result which did the division of the Rhine length of input data in the Rhine length of output data, and counts them using an adder, and since the data-processing section operates so that the input data memorized by the input buffer may be chosen in order and it may consider as output data, enlarging-or-contracting processing by hardware can be performed automatically continuously. Therefore, it becomes possible to perform this kind of image processing at a high speed, without giving a load to CPU.

[0041] Moreover, with above equipment, in order to perform processing in which read the input data memorized by RAM4 and a final processing result is written in RAM4, there are few pulse duty factors of a bus and other processings by CPU are not barred. That is, it becomes unnecessary to repeat RAM4 and to access frequently for input data processing, and the count of total access of RAM4 can fully be reduced.

[0042] <Example 2> By the example 1, processing for reading input data is performed at once from RAM4 in the processing for generating print data from the input data memorized by RAM4, and the occupancy time amount of a bus line was shortened. Here, after the above enlarging-or-contracting processings, if it is going to perform binarization processing of the data further, the procedure which carries out binarization processing of the middle data again memorized by RAM4, and is used as print data must be performed. This

makes the count of access to RAM4 increased like processing by the conventional CPU. So, by this example, the data-processing section 10 also performs binarization processing to enlarging-or-contracting processing and coincidence by one read-out.

[0043] The data-processing section block diagram of an example 2 is shown in drawing 6. This data-processing section is equipped with the binarization circuit 30 and the dither data circuit 31 other than an input buffer 11, the data selection circuitry 12, the enlarging-or-contracting counter 13, and an output buffer 14. Namely, it differs in that the binarization circuit 30 and the dither data circuit 31 were newly added to the circuit of an example 1.

[0044] The dither data circuit 31 consists of the dither data readout circuitry 33 and the dither data memory 34. The dither data 35 are memorized by the dither data memory 34. This dither data 35 is written in the dither data memory 34 by CPU1 shown in drawing 2. In addition, when there is much amount of data of dither data, as the block of the broken line of drawing showed, the DMA circuit 32 for dither data automatic transfer is used.

[0045] For example, the amount of 1 pixel always makes the input buffer 11 the configuration which memorizes the gradation data which are 8 bits of input data. Even if the input data which follows, for example, is actually read from RAM4 is expressed by 1 pixel 2 bits or 1 pixel 4 bits, it will normalize to 1 pixel 8 bits here. In this way, binarization processing of the 1-pixel 8-bit data for one line held at the input buffer 11 is carried out by the binarization circuit 30.

[0046] Concrete actuation of the above-mentioned binarization circuit 30 and the dither data circuit 31 is explained. Every 1 pixel of 8-bit gradation data which the data selection circuitry 12 outputs is sent into the binarization circuit 30 in order. The binarization circuit 30 measures the threshold outputted from the dither data circuit 31, and the output of the data selection circuitry 12, and performs binarization processing. That is, when the output of the data selection circuitry 12 is larger than a threshold, it is referred to as "1", and when other, the output "0" is written in an output buffer 14.

[0047] The above-mentioned dither data can consist of threshold groups of a large number required in order to process the pixel for one line outputted from the data selection circuitry 12. The dither data readout circuitry 33 operates as it returns to a head again and the corresponding threshold is read, after reading a threshold group from the dither data memory 34 in order and reading from a head to the last. Binarization of the data which the data selection circuitry 12 outputs by this using the threshold of a large number designed intricately can be performed.

[0048] What is necessary is just to set the counter for read-out of a threshold so that a threshold may be again supplied from a head when preparing the threshold for one line, supplying the threshold corresponding to order from the head in each Rhine as for example, dither data and moving to next Rhine in supplying the threshold of one piece to the data for 1 pixel, respectively. Moreover, in the case of the dither pattern of 4x4 matrices, the threshold for four lines is prepared, for example. It carries out as each threshold in dither data is read in order from the 1st line to the 4th line and the threshold of the head in dither data is again read at the head of the 5th line.

[0049] For example, in adopting the dither pattern of 16x16 matrices, the amount of data of dither data becomes 16 lines. It is uneconomical to prepare into this memory which stores such all a lot of thresholds. In this case, it is good to store dither data in RAM4 and to prepare a Direct-Memory-Access circuit as shown in the broken line of drawing 6 which divides the specified quantity every and is accepted to predetermined timing. The threshold of four pieces is supplied in order to the data for 1 pixel, and control which outputs four binarization data can also be performed. What is necessary is just to increase the data transfer clock after binarization processing 4 times to a binarization processing earlier data transfer clock.

[0050] Moreover, if a threshold is read so that the threshold of one piece may be made to correspond to one gradation data of each pixel outputted from the data selection circuitry 12, binarization of the 8-bit gradation data will be carried out as it is, for example, and a 1-bit printing image will be obtained. That is, the image data for one line held at the input buffer 11 has the amount of data reduced by 1/8, and is stored in an output buffer 14. Even when the Rhine length is expanded twice, the data of the quadrant of an input buffer are stored in an output buffer. When the Rhine length is set to 1/2, 1/16 of the data of an input buffer are stored in an output buffer 14.

[0051] <Effectiveness of an example 2> According to this example, as mentioned above, that to which

enlarging-or-contracting processing was ended, binarization processing was ended, and the amount of data fully decreased is written in RAM4 shown in drawing 1. And the data is the print data which can perform printing processing immediately. Input data is repeatedly read from such a thing like before, data processing is performed, and the count of data read-out writing to RAM4 can be sharply reduced compared with the case where it is said that print data are finally obtained.

[0052] <Example 3> When expand as mentioned above for the scale factor which had the input data specified, it reduces or binarization processing is carried out, there is a case so that some output data may cross the field which can be printed. It is necessary to remove in such a case for a part of trailer [a part for the head part of input data, or a trailer, and head part of output data]. It becomes possible by adding the Delete function of such garbage data to the above-mentioned data-processing section to mitigate the burden of CPU further. Such a function is realized after this example 3.

[0053] Drawing 7 is the data-processing section block diagram of an example 3. The offset control circuit 40 established in the input side of the input buffer 11 of this drawing is newly added by this example. Since the other structure and the other function of a circuit may be the same as that of the thing of an example 2, they omit explanation.

[0054] In order to explain actuation of the circuit of drawing 7, and actuation of the offset control circuit 40, the contents of data were displayed on the location surrounded by the alternate long and short dash line of the drawing 7 lower left part. This data-processing section shall have the function shown in drawing to expand the input data D1 for one line, for example, to carry out binarization processing, and to obtain the output data D0 for one line. Here, suppose that garbage data DD exists in the head part of input data D1. With garbage data, when the following drawing data lap with front drawing data in the case where the painting-out section by the drawing function is for example, in former data, a lap part serves as unnecessary data for the following drawing data. This garbage data DD intercepts the first data for 8 pixels of the input data transmitted every 1 pixel from the DMA circuit 8 which showed the offset control circuit 40 to drawing 1 when it was by 8 pixels, and the transfer to an input buffer 11 is prevented.

[0055] Such a cutoff circuit can consider various kinds of things. For example, the gate circuit which carries out a mask by 8 pixels can constitute the register which holds the data for 1 pixel temporarily, and the transfer clock which transmits the data which held to this register to an input buffer 11 from the head of the data for one line. moreover -- or the gate control circuit which only the transfer time for 8 pixels intercepts can constitute the gate from the head of the data for one line with the gate group inserted all over the circuit which transmits the data for 1 pixel to an input buffer.

[0056] Moreover, when 8 pixels of heads of the 512-pixel input data consider as garbage data, for example, the shift register which can hold the data for 504 pixels is used as an input buffer 11. When 512-pixel data were transmitted to this input buffer 11 in order and the data of the last pixel are written in, the OPA flow of the first data for 8 pixels is carried out, and they are cut off. Moreover, there is also the approach of using an input buffer 11 as the usual memory. In this case, all input data is first written in an input buffer. The data selection circuitry 12 memorizes the number of offset which CPU specified. And only the number of offset is read and an initiation start address is shifted.

[0057] Two examples of the beginning are examples which arrange the offset control circuit 40 just before an input buffer 11. Moreover, the two remaining examples are examples which constituted the input buffer 11 and the offset control circuit 40 in one. Any example is realizable using the easy hardware for data processing known well from the former. In other examples of explaining below, such a circuit can be used similarly.

[0058] <Effectiveness of an example 3> It becomes possible to generate the print data which adjust the Rhine length of output data and carry out expansion or contraction processing as it is and which can be transmitted to a printing control section by forming the offset control circuit 40 which has the function to delete the data of the predetermined number of the head part of input data, as mentioned above. Thereby, the burden of CPU is mitigable.

[0059] <Example 4> The above-mentioned example explained the example which deletes the head part of input data. In the case of this example 4, the garbage of the head of output data is deleted. The data-processing section block diagram of an example 4 is shown in drawing 8. As shown in this drawing, the offset control circuit 41 is formed between the binarization circuit 30 and the output buffer 14. Parts other

than this offset control circuit 41 are good with the completely same configuration as the circuit of an example 2. Therefore, explanation of other parts is omitted.

[0060] The offset control circuit 41 has the function to delete garbage data DD of the predetermined length of the head part of the output data D0 transmitted from the binarization circuit 30, as shown in the lower left side of drawing. The relation between the offset control circuit 41 and an output buffer 14 is the same as the offset control circuit 40 of an example 3, and the relation of an input buffer 11. Therefore, the offset control circuit 41 is realizable with the same hardware top as an example 3.

[0061] <Effectiveness of an example 4> As mentioned above, if the head part of the data after binarization is automatically deleted at the time of output-buffer storing, it will become possible to process as print data as it is.

[0062] <an example 5> -- the above -- when processing the image data for 1 page, any example can do processing efficiently, if the data for at least one line are held and processed to an input buffer. However, in addition to this, the data for one line may be unable to be collectively stored in an input buffer neither according to the capacity of an input buffer, nor various situations. In this case, data of one line may be divided into every [1/several], it may transmit to an input buffer, and enlarging-or-contracting processing may be carried out separately. By this example, the print-data length in such a case is adjusted.

[0063] The data-processing section block diagram of an example 5 is shown in drawing 9. The part except the Rhine length amendment circuit 42 of this drawing is the same as that of the circuit of an example 4. The Rhine length amendment circuit 42 has the function to always adjust the Rhine length of the data written in an output buffer 14 to fixed die length. If the data of the Rhine length L decided beforehand are specifically received as shown in the right-hand side of drawing, the surplus data beyond it will be canceled.

[0064] On the other hand, when the Rhine length of the data written in an output buffer 14 does not fulfill the above-mentioned Rhine length L, only the number of bits which ran short fills up data. The bit of the tail of the written-in data is copied as the concrete data supplement approach, and it is based on the approach of adding only a required number.

[0065] The Rhine length amendment circuit 42 is equipped with the register which memorizes the data equivalent to for example, the Rhine length L. The data of the predetermined length after binarization processing input every 1 pixel into the offset control circuit 41. An output buffer 14 receives every 1 pixel of data inputted from the offset control circuit 41, and writes them in a predetermined field. At this time, the output of the offset control circuit 41 is incorporated by the Rhine length memorized in the Rhine length amendment circuit 42. Therefore, even if the data more than the die length specified by the Rhine length amendment circuit 42 from the offset control circuit 41 input into an output buffer 14, surplus data are not written in an output buffer 14.

[0066] On the other hand, when the data length outputted from the offset control circuit 41 is shorter than the data length memorized in the Rhine length amendment circuit 42, the offset control circuit 41 suspends actuation, with the data equivalent to the last pixel held. An output buffer 14 continues reading of the data which the offset control circuit 41 held, in order to receive the data for the Rhine length from the offset control circuit 41. That is, the same data are repeated and read until it reaches the Rhine length L. In this way, a supplement of insufficient data is attained.

[0067] The explanatory view of the data-processing section of operation is shown in drawing 9. According to the equipment of the above-mentioned example, the following actuation is realized. (From a) to (c) is the explanatory view. When performing processing which obtains the output data for one line of the Rhine length L as shown in (a) of drawing, it considers dividing and processing input data as mentioned above. For example, if it processes by dividing into L2 in L1 and the second half of the first half, both will be connected and the output data for one line of the Rhine length L will be obtained. However, as enlarging-or-contracting processing and binarization processing show to (b) of drawing, only as for G, the 1st processing wants data for L1, supposing the 2nd processing wants data for L2 only as for G, a clearance will be generated in output data and image quality will deteriorate.

[0068] On the other hand, when die length of R increased from the die length of L1 and L2 and it collects into one line, as shown in (c), the lap for R minutes will arise the first half and the second half also. This also causes image quality degradation. The Rhine length amendment circuit 42 of the data-processing section shown in drawing 9 adjusts the Rhine length of data which stores in an output buffer 14

automatically. In addition, although each of L1 and L2 should just be 2/L, both may differ.

[0069] <Effectiveness of an example 5> Since data are written in an output buffer 14, also when a data length becomes irregular by enlarging-or-contracting processing or binarization processing as mentioned above, adjusting a data length using the Rhine length amendment circuit 42, the output data arranged with fixed die length are obtained. By this, processing of a consecutiveness circuit becomes easy, and also degradation of image quality can be prevented.

[0070] <Example 6> For example, when performing contraction processing extreme like 1/100 and 1/1000 for the inputted image, as for data required for the data generation after contraction processing, former data are no longer a part very much. That is, processing will be performed, picking up the data with which 1000 of former data are dotted at intervals of one piece. Therefore, all the input data memorized by RAM4 is transmitted to the data-processing section, and it becomes the futility of a storage region to perform the operation for contraction processing after that. So, by this example, a skip function is added to the DMA circuit 8 shown in drawing 1.

[0071] Drawing 11 is the print-data generation equipment block diagram of an example 6. The skip control circuit 43 is connected to the DMA circuit 8 of the equipment of drawing. There is no place which other circuits change with the thing of an example 5. In addition, about other circuits, it cares about by one or less-example neither of the configurations. Here, drawing 10 is referred to again. What was shown in (d) of drawing 10 is input data memorized by RAM4. The number was given to the data of each pixel like "0", "1", "2", "3", and -- in order.

[0072] For example, when it turns out beforehand that such data are reduced to 1/20, the above-mentioned DMA circuit 8 skips every 20 pixels of data in RAM4, and reads them, and it operates so that it may transmit to an input buffer 11. In drawing, the data to read were surrounded and illustrated by the frame of a thick line. Now, the storage region of an input buffer 11 can also be reduced enough, and the count of access to RAM4 by the DMA circuit 8 can also be decreased.

[0073] CPU1 shown in drawing 2 sets reduction percentage called 20 minute 1 as the enlarging-or-contracting counter 13 shown in drawing 11. The skip control circuit 43 sets the start address in consideration of reduction percentage, the amount of skips, and the stop address to the DMA circuit 8. Thereby, the DMA circuit 8 transmits only data required for the enlarging-or-contracting processing after opposite *Perilla frutescens* (L.) Britton var. *crispa* (Thunb.) Decne. to an input buffer 11. In addition, actuation of this example is effective when the reduction percentage of data is remarkable. Control may become [the way where it follows, for example, reduction percentage does not operate the skip control circuit 43 by 1/2 and about 1/3] easy. In this case, you may make it set the information on whether CPU1 confirms skip control to the skip control circuit 43, or it does not carry out. What is necessary is just to write in the criteria of decision that it is better that it is better whether to carry out effective [of the skip function], and whether to make it an invalid, into the program of CPU1.

[0074] In addition, in the data-processing circuit explained in the example 1 grade, the enlarging-or-contracting counter 13 has calculated the address of the data memorized by the input buffer 11. The data selection circuitry 12 chooses the data in an input buffer 11 using this address. Therefore, in order to perform the same actuation as the example 1 grade explained, it is desirable that all the input data memorized by RAM4 is written in in the address as it is into an input buffer 11.

[0075] In this case, what is necessary is just to write the dummy data of the 2nd - the 19th part in an input buffer 11, by the time the DMA circuit 8 transmits the data of the 20th ["0" / "20th"] watch to an input buffer 11, after transmitting the data of eye watch. What is necessary is just to carry out write-in actuation of an input buffer 11 to this with a control clock 20 times the rate of the DMA circuit 8. The contents of dummy data are arbitrary. Even in this case, the count of access of RAM by the DMA circuit 8 is decreased, and it is effective in reducing bus occupancy time amount.

[0076] Of course, the DMA circuit 8 can also write in order the data which carried out skip read-out in an input buffer 11 as it is. In this case, since contraction processing of input data is already completed, the enlarging-or-contracting counter 13 does not operate. The data selection circuitry 12 should just output the data written in the input buffer 11 to the binarization circuit 30 as it is.

[0077] <Effectiveness of an example 6> As mentioned above, since the DMA circuit 8 performs a data transfer from RAM4 to an input buffer 11 by skip read-out, compared with the case where all data are

transmitted to an input buffer 11 from RAM4, it becomes possible to shorten the data read-out processing time by the DMA circuit 8. Moreover, it becomes possible to shorten the time amount which occupies the bus of CPU by this.

[Translation done.]

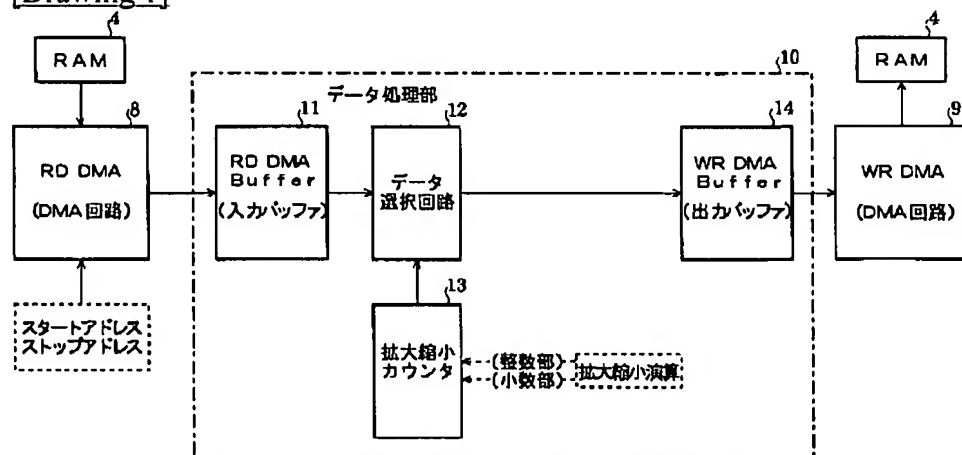
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

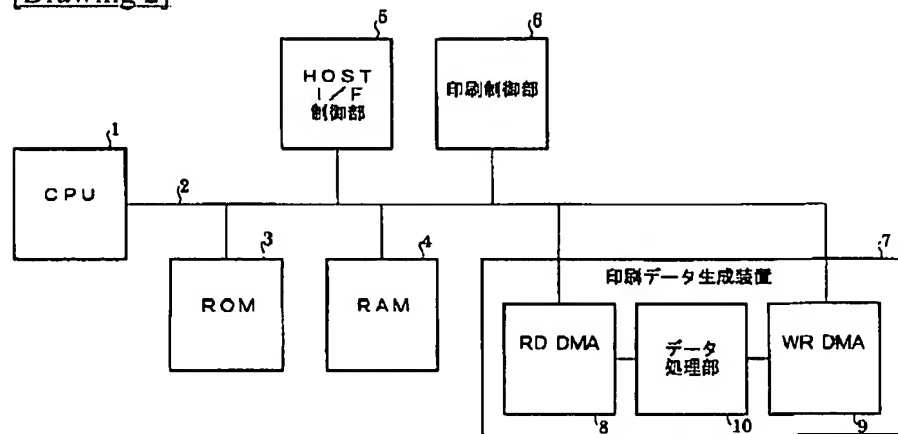
DRAWINGS

[Drawing 1]



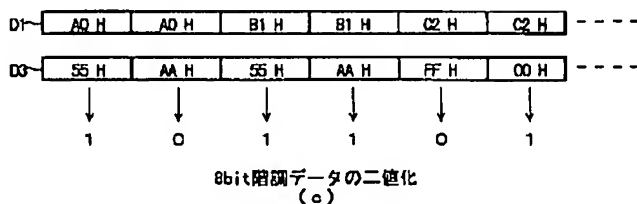
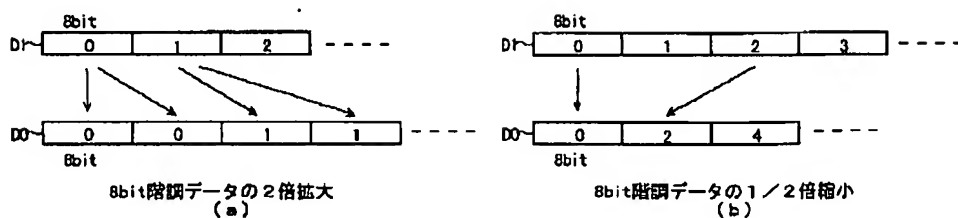
具体例1の印刷データ生成装置ブロック図

[Drawing 2]



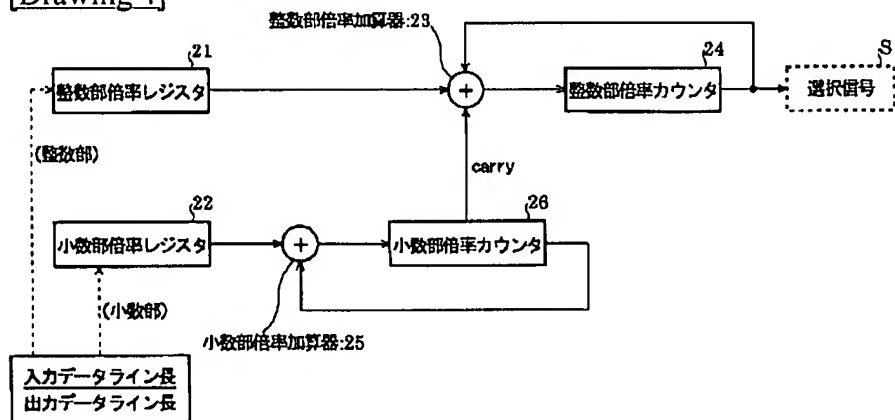
印刷装置のブロック図

[Drawing 3]



データ処理部の機能説明図

[Drawing 4]



拡大縮小カウンタのブロック図

[Drawing 5]

出力データ 画素位置	1/2倍縮小 整数部=2 小数部=0.0	1/3倍縮小 整数部=3 小数部=0.0	1/4倍縮小 整数部=4 小数部=0.0	1/5倍縮小 整数部=5 小数部=0.0
	R reg : Preg	R reg : Preg	R reg : Preg	R reg : Preg
0	0.0	0.0	0.0	0.0
1	0.0	2	0.0	4
2	0.0	4	0.0	8
3	0.0	6	0.0	12
4	0.0	8	0.0	16
5	0.0	10	0.0	20

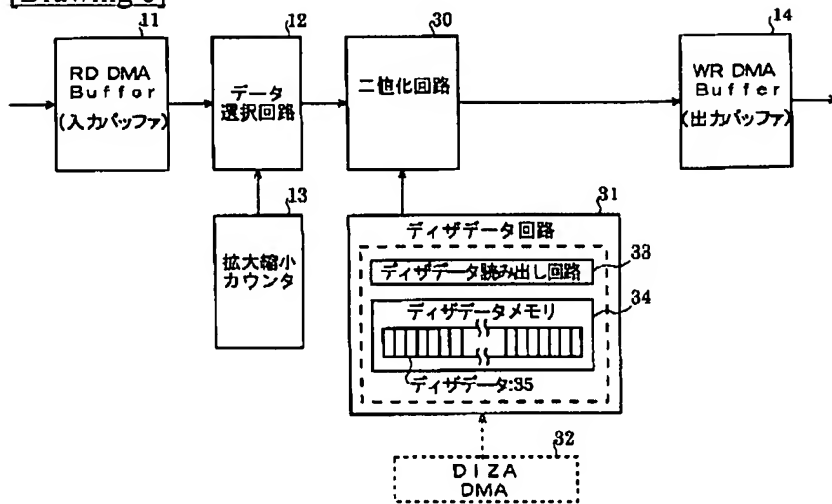
(a)

出力データ 画素位置	2倍拡大 整数部=0 小数部=0.50	3/2倍拡大 整数部=0 小数部=0.66	5/2倍拡大 整数部=0 小数部=0.40
	R reg : Preg	R reg : Preg	R reg : Preg
0	0.00	0	0.00
1	0.50	0	0.40
2	1.00(-0.00)	1	0.80
3	0.50	1	1.20(-0.20)
4	1.00(-0.00)	2	0.80
5	0.50	2	1.00(-0.00)
6	1.00(-0.00)	3	0.40
7	0.50	3	0.80
8	1.00(-0.00)	4	1.20(-0.20)
9	0.50	4	0.60
10	1.00(-0.00)	5	1.00(-0.00)

(b)

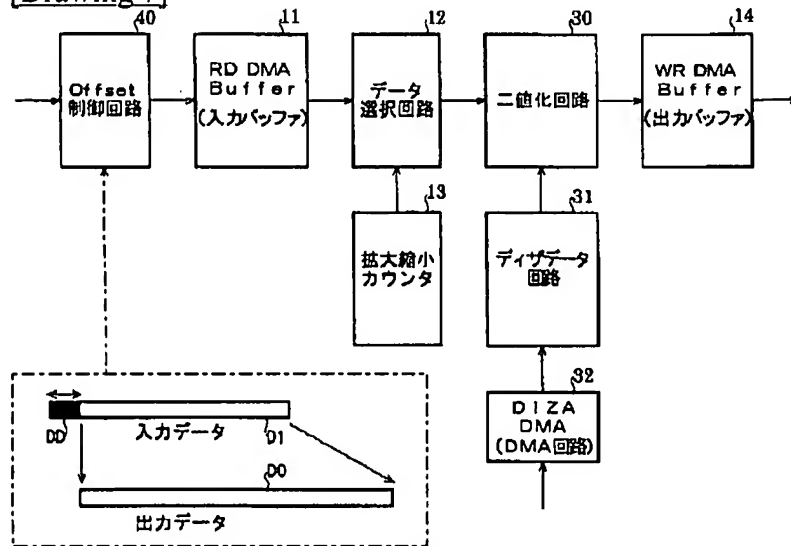
拡大縮小動作説明図

[Drawing 6]



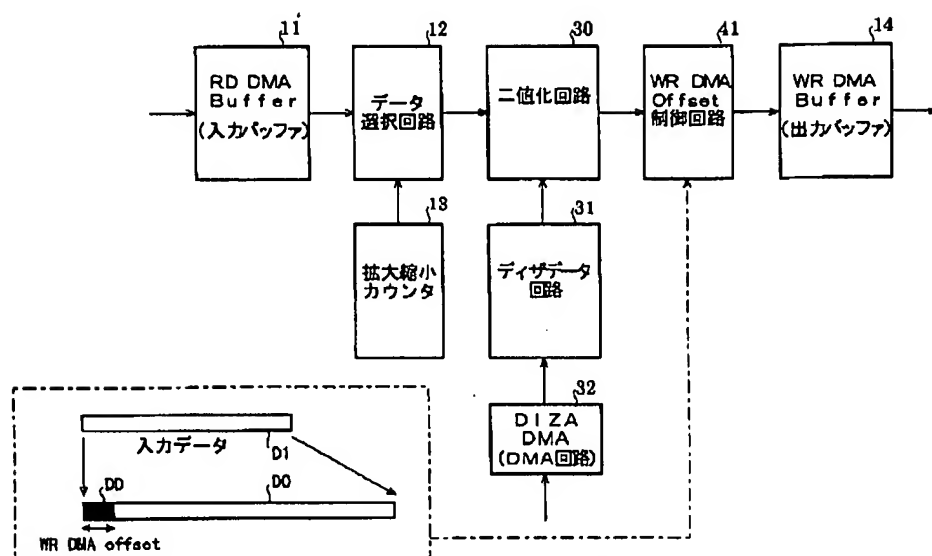
具体例 2 のデータ処理部ブロック図

[Drawing 7]



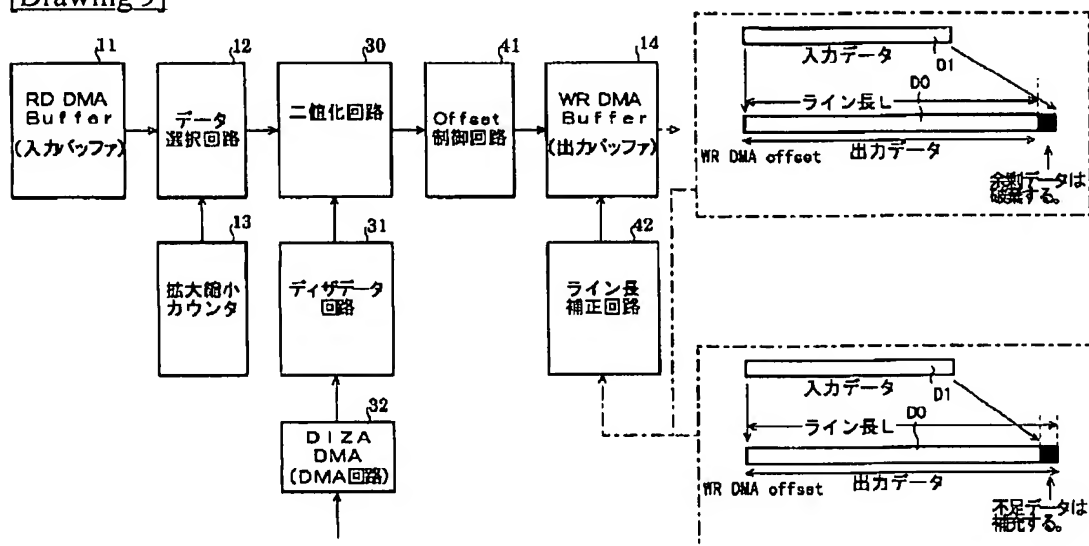
具体例 3 のデータ処理部ブロック図

[Drawing 8]



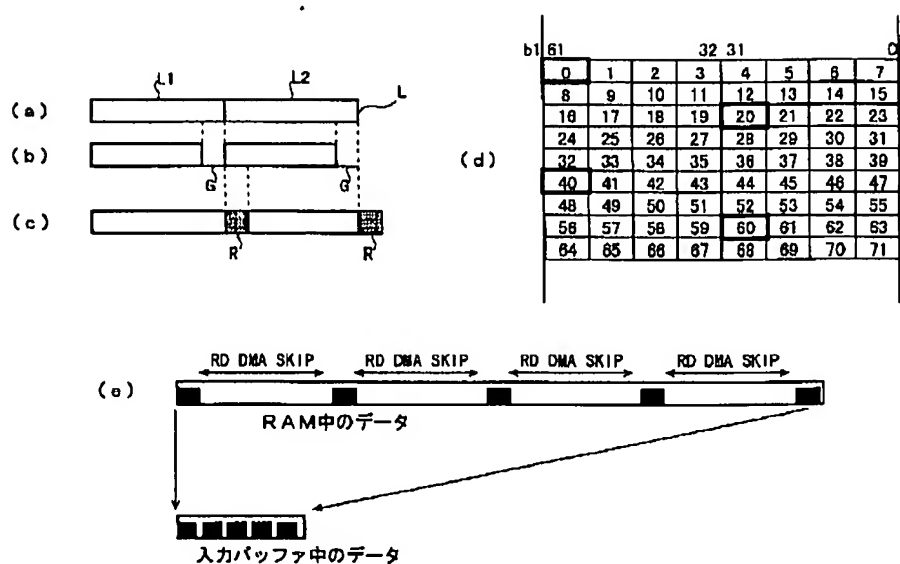
具体例 4 のデータ処理部ブロック図

[Drawing 9]



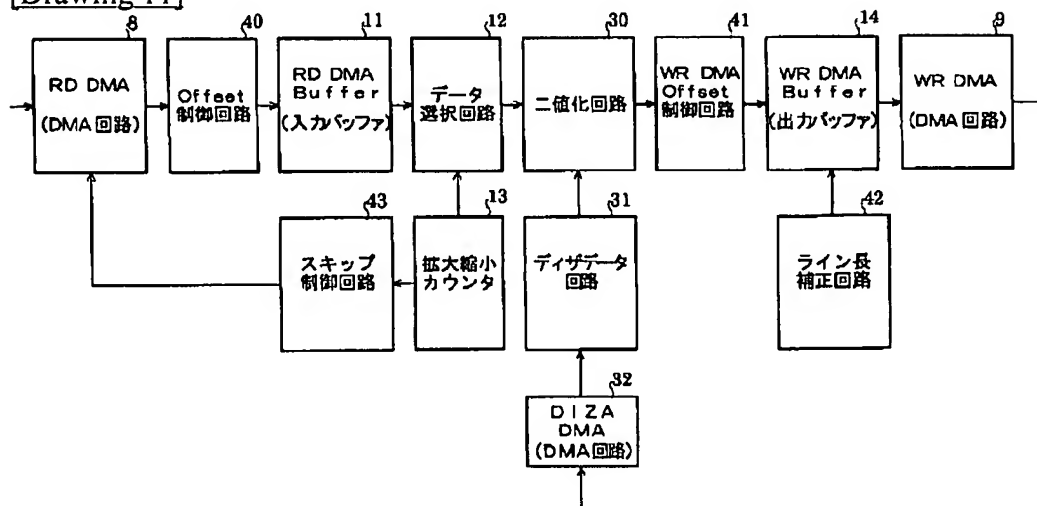
具体例 6 のデータ処理部ブロック図

[Drawing 10]



データ処理部の動作説明図

[Drawing 11]



具体例6の印刷データ生成装置ブロック図

[Translation done.]